Master's thesis

Automatic classification of segmented seismic recordings at the Nevado del Ruiz volcano, Columbia

By
Remco Hoogenboezem

Supervisor: Dr. Ir. R. Duin

Abstract

The Nevado del Ruiz volcano is an active and dangerous volcano in the Andean volcanic belt. Measuring seismic activity is one of the most reliable and widely used techniques to monitor and predict renewed volcanic activity. Seismic activity can be caused by several different underlying physical processes. It is of interest to the earth-science observatories monitoring potentially dangerous volcanoes to determine the underlying cause of the registered earthquakes. Typically segmented seismic recordings are classified by hand often based upon their frequency contents. An automated system capable of discriminating reliably between several different seismic recording classes can potentially release the human expert from the labor intensive classification task. An Interesting question concerning the frequency representation of the segmented seismic recordings is: if it is better to use only frequency information in the form of a single spectrum or to use a time frequency representation such as a spectrogram. Furthermore it is of interest to see if the ordering of the spectral frames inside the resulting spectrograms is of importance. In this study a justified spectrogram representation is developed for the segmented recordings from the Nevado del Ruiz volcano. Using this spectrogram representation we also look at five different classification strategies in combination with a large number of different classifiers. Often seismic events such as volcanic tectonic earthquakes, tectonic earthquakes, rockfall etc... are registered by several seismic stations. It is of interest to see if the recordings of multiple stations can be combined to improve classification results. Furthermore it is of interest to see how well the untrained and trained classifier systems generalize to the recordings of other stations.

Contents

8 Conclusion and recommendations

References

1 Introduction

1.1 The Nevado del Ruiz volcano

The Nevado del Ruiz volcano is an active stratovolcano. A stratovolcano is a tall conical shaped volcano. Stratovolcanoes are characterized by gentle lower slopes but steep upper slopes. Furthermore stratovolcanoes usually have a narrow summit crater. Stratovolcanoes are composed of many layers (strata) of hardened lava, volcanic ash and other volcanic material. Because of their composite layered structure stratovolcanoes are sometimes also referred to as composite volcanoes.

Figure 1.1: Schematic representation of a stratovolcano (left). The Nevado del Ruiz volcano (right).

The Nevado del Ruiz volcano is the northernmost of several Colombian stratovolcanoes in the Andean volcanic belt. The Andean volcanic belt is the result of subduction of the Nazca plate beneath the South American continental plate. Subduction is a process that occurs when two tectonic plates move towards each other. One of the two tectonic plates moves over the other tectonic plate causing the second tectonic plate to sink into the earths mantle. Oceanic tectonic plates are heavier compared to continental tectonic plates. Therefore when a oceanic tectonic plate collides with a continental plate, the oceanic plate sinks into the earths mantle. Subduction is a process that is typically measured in centimeters per year. The subduction velocity of the Nazca plate with the South American continental plate is in the order of nine centimeters per year. A subduction zone is the area were two tectonic plates meet and were subduction occurs. Subduction zones are often characterized by high volcanic activity and frequent occurrence of earthquakes.

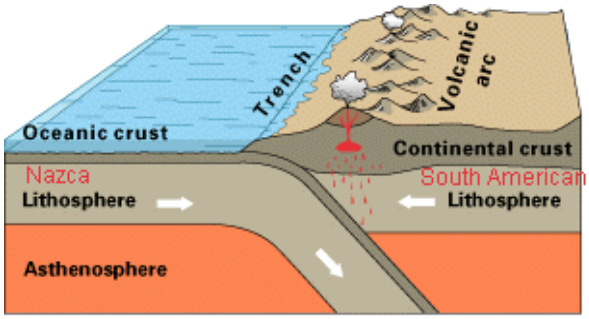

Figure 1.2: Schematic representation of the subduction process

The Nevado del Ruiz volcano is a very tall volcano. It's summit reaches 5389 meters above sea level. Although the volcano is located approximately 500 kilometers from the earths equator its summit is covered with 25 square kilometers of snow and ice. This is also were part of the name of the volcano comes from. Nevado means snowy or snow-capped.

Stratovolcanoes often produce highly explosive and dangerous eruptions. This type of eruption is also known as Plinian. This type of explosive eruptions are called after Pliny the Younger. Pliny the Younger witnessed the famous eruption of the Vesuvius in 79AD and wrote a remarkable objective account on the eruption. Plinian eruptions are characterized by a large column of gas and other volcanic material (mostly pumice) that is emitted by the volcano at incredible force. The column often reaches high into the stratosphere. The deposit resulting from this column often covers large area's. During a Plinian eruption the volcano can also emit a large amount of magma. Sometimes the emission of magma is so large that the crater collapses. The resulting volcano is referred to as a caldera volcano. Plinian eruptions can also produce pyroclastic flows. A pyroclastic flow is a fast moving current of hot gasses and other volcanic material (not lava). The velocity of a pyroclastic flow can be as high as 700 kilometers per hour. The velocity of the pyroclastic flow depends on the amount of gas that is emitted per unit of time, the density of the material inside the flow and the gradient of the volcano. Usually pyroclastic flows travel close to the ground. The temperature inside a pyroclastic flow can reach 1000 degrees Celsius. The combination of its high velocity, high temperature and close to ground move pattern makes the pyroclastic flow very dangerous. It is estimated that during the famous eruption of the Vesuvius in 79AD, 62 percent of casualties inside Pompeii were caused by pyroclastic flows. Most other casualties were caused by collapsing buildings. The duration of a Plinian eruption can vary between less than a day up to several months.

Figure 1.3: A rendering of how the 79AD Vesuvius eruption might have looked like. (left) An example of a pyroclastic flow (right)

The most recent eruption of the Nevado del Ruiz volcano was on 13 November 1985. This was the third eruption of the volcano in 400 years. Prior to the eruption the volcano had been active for almost a year, producing minor earthquakes and steam explosions. The eruption began at 3:06 pm. Large amounts of pumice and ash were emitted into the air. Two hours later the deposit of the ash cloud reached the city of Armero. Armero is a medium sized city with over 28000 residents. At 7:00 pm the red cross ordered an evacuation of the town. But shortly after the evacuation was ordered the ash and pumice stopped falling and the evacuation was canceled. At 9:08 pm the volcano resumed its eruption. This time molten rock was violently emitted by the volcano. Furthermore the volcano produced pyroclastic flows. The pyroclastic flows began to melt the summit ice cap. The molten ice in combination with the pyroclastic flows caused several lahars. A lahar is a volcanic mudflow

often composed of water, volcanic ash pumice, and clotted lava. One of the lahars followed the river Cauca and submerged the village Chinchina. The village was completely destroyed and 1927 people were killed. Other lahars followed the path of previous mudflows caused by previous eruptions. The largest of the lahars reached the city of Armero. Most of the buildings were destroyed and buried in a matter of minutes. 21000 people were killed in the city of Armero. Some people were killed after the eruption caused by their injuries or due to infection. In total 23000 people were killed and another 5000 people were injured. The eruption of November 1985 was the second deadliest eruption of the 20eith century. The eruption of Mont Pelée (Martinique) was even worse in terms of the number of casualties. New houses were build by the government for those who survived the disaster. It is estimated that the 1985 eruption cost the Colombian government 7,7 billion dollars which is approximately 20 percent of the annual Colombian national product.

The high number of casualties can be partly explained by the fact that most people were unaware of the pyroclastic eruption due to bad weather conditions at the summit. Prior to the 1985 eruption Colombian volcanoes were not daily and sufficiently monitored. Furthermore their were reassuring messages from the mayor via the radio and from a local priest over the church public address system. Most people did not belief that the volcano was about to erupt violently. When the city of Armero was build the local authorities ignored the fact that the location of the new city has a high mudflow risk. The city of Armero was build on the remainders of previous mudflows.

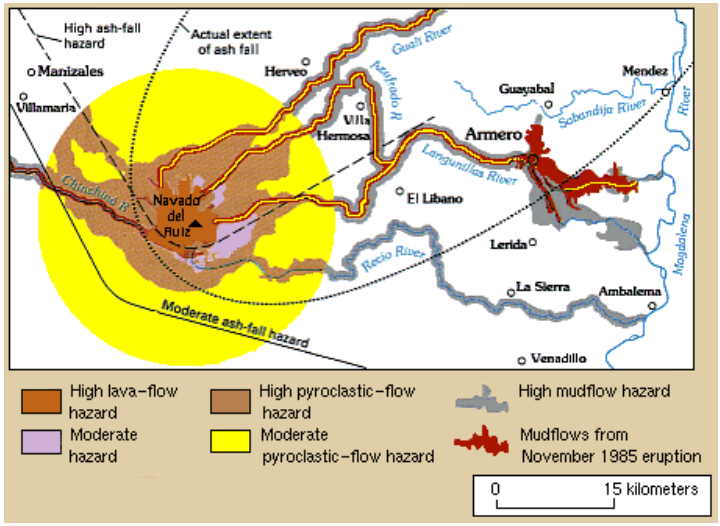

Figure 1.4: The Nevado del Ruiz hazard map

Figure 1.5: The deadly lahar destroys the city of Armero.

2 Problem description

2.1 Introduction

As a response to the 1985 eruption the Nevado del Ruiz volcano and four other potentially dangerous volcanoes are monitored by the Volcanological and Seismological Observatory at Manizales (VSOM Abbreviated). Seismic activity can be an important indicator for renewed volcanic activity. Therefore seismic activity is measured by several strategically placed seismic stations. The resulting digital measurements are send to the observatory for evaluation. See figure 2.1

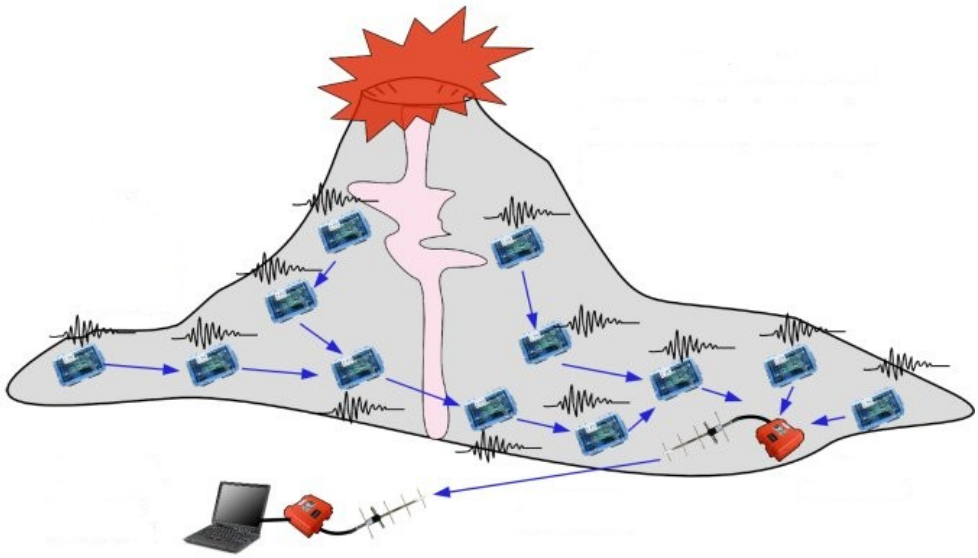

Figure 2.1: Earth crust movement is measured at several seismic stations and send to the observatory

Earth crust movement can have several different causes. It is of interest to the VSOM staff to determine the cause of an incoming seismic signal. For example is a given seismic signal the result of transport of magma in the earths crust? Or was it just caused by rockfall?

Regions of interest in the seismic recordings are still classified by hand by the VSOM staff. This is a time consuming and labor intensive procedure. An automated system could be of great help releasing the human experts from the task of classifying regions of interest into the appropriate seismic event class. Finding regions of interest in a much longer seismic recording is not difficult because most earthquakes have a much higher amplitude compared to the neighboring background or noise signal. One could for example compare the current average signal amplitude with the neighboring average signal amplitude followed by a thresholding scheme. [5]

This study is focused on classifying regions of interest into a subset of seismic signal classes using a pattern classifier. Thus in this study one is not interested in segmenting a region of interest from a much longer seismic recording. The regions of interest are given. A simplified pattern classifier according to [3] is given in figure 2.2. The pattern classifier consists of a sensor block, a feature selection or extraction block and a classifier block. In this study the sensor block is a digital seismometer producing a continuous seismic recording followed by a segmentation stage producing only regions of interest from the continuous recording. The sensor block is given. The sensor representation is a region of interest. The feature selection/extraction block is responsible for reducing the dimensionality of the sensor representation and providing a relevant set of features for the classifier block. Ideally the feature selection/extraction block removes all redundant information but maintains all information that contributes to classification performance. The feature

representation is the result of applying the feature selection/extraction block on the sensor representation. Finally the classifier block predicts the event type based upon the given feature representation. Several possible solutions for the feature selection/extraction and classifier blocks are discussed and tested in this study.

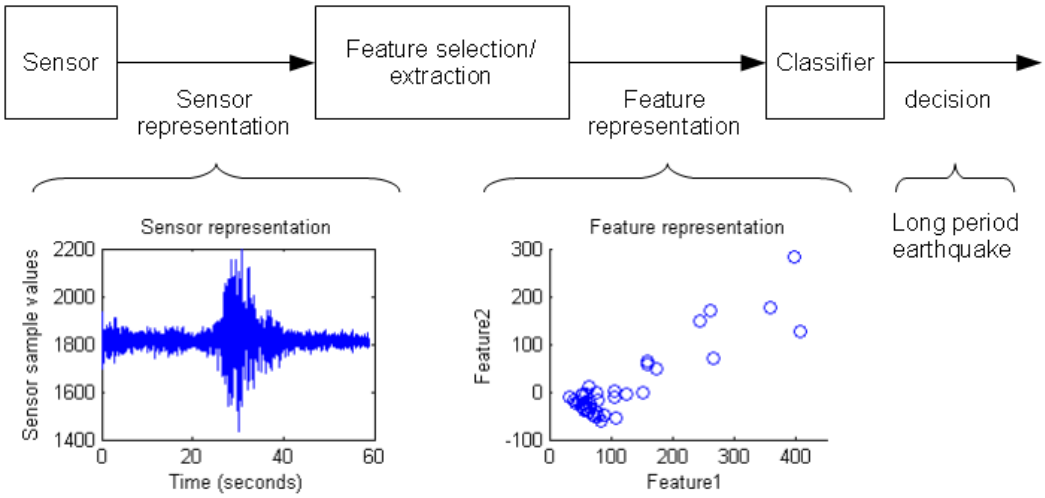

Figure 2.2: Pattern classifier according to [3]

When classification of regions of interest is done by hand often frequency information is used in the form of a single spectrum. A single spectrum only contains frequency information, all time information is lost. Often different underlying physical causes of earth crust movement produce different frequencies and or combinations of frequencies. In other studies frequency information was already successfully used to discriminate between several seismic event classes. Therefore also in this study frequency information is used to discriminate between different seismic event classes. The typical segmented seismic recording is not stationary. Meaning that signal characteristics such as amplitude (see for example figure 2.2) and frequency vary in time. Furthermore the typical segmented seismic recording is the result of registering several different wave types. Several different often occurring wave types will be discussed in chapter 3.

Fundamental questions concerning the representation of the segmented seismic recordings are:

- Can one discriminate automatically and successfully between different seismic recordings using only a single spectrum? Or can one achieve better classification results using a time frequency representation such as a spectrogram?
- Whilst using the spectrogram representation is the ordering of the spectral frames of importance? If so how can one model the ordering successfully?

Questions of interest concerning the recordings of multiple stations are:

- Can one improve classification results by combining recordings from several different seismic stations?
- How well does the untrained and trained pattern classifier generalize to the recordings of other seismic stations?

3 Seismic waves (definition, medium and registration)

3.1 Seismic waves

3.1.1 Waves

What are seismic waves? A wave is often understood intuitively as the transport of a disturbance in space. It is tempting to say that waves always travel through a medium. But this statement is falsified by the fact that electromagnetic waves can travel through a vacuum. Waves travel and transfer energy, often with no permanent displacement of the particles in the medium. It is possible that a wave travels from one point to another without the displacement of mass. But again there is an exception. For a standing wave it is difficult to say that it is moving from one point to the other. A standing wave can occur because the medium is moving in the opposite direction (at equal propagation velocity) compared to the wave. A standing wave can also be the result of two interfering waves traveling in opposite direction.

3.1.2 Mechanical waves

A mechanical wave propagates or travels through a medium thanks to the restoring force of the medium. The restoring force tries to maintain an equilibrium in the medium. When the equilibrium is disturbed for example by local deformation of the medium, the restoring force tries to bring back the equilibrium inside the medium. Often the restoring force over compensates in response to a disturbance, causing a overshoot past the equilibrium. The result is an oscillating medium. A mechanical wave requires an initial energy input (the disturbance). Once the initial energy is added the wave will travel through the medium until all the initial energy is dissipated. Transport of the disturbance in the form of a mechanical wave costs energy. The propagation velocity of the resulting wave depends on the elasticity and density of the material. The final propagation distance depends on the amount of initial energy and also on the elasticity and density of the material.

3.1.3 Seismic waves

A Seismic wave is an example of a mechanical wave. The medium of seismic waves is mostly the earth. But seismic waves can also propagate through water and unfortunately through man made structures. Some seismic waves (P-waves) can travel through the air and are audible. Seismic waves just like other mechanical waves are caused by a disturbance of the medium. Often the disturbance is a sudden release of build up energy in the earths crust. But there are also other causes for seismic waves. The location inside the earths crust where most of the energy is released is also referred to as the hypocentrum. The location on the earths surface where the amount of remaining energy is highest is called the epicentrum.

3.1.4 Seismic wave types

Seismic waves can be roughly categorized into two types of waves. The categorization is based upon the way the seismic waves travel through the medium. Note that the enumeration of seismic wave types is not complete but the most important types are given. The first type of seismic waves are referred to as body waves. The second type of waves are called surface waves. Body waves can travel through the earths inner layers. Whereas surface waves like the name already suggests can only travel along the earths surface. Earthquakes usually radiate both body waves and surface waves. Surface waves are almost always responsible for most of the damage in the event of an earthquake. Because surface waves can only travel along the earths surface the energy of surface waves is reduced for deeper earthquakes. But because surface waves radiate energy only in two dimensions energy decline is slower compared to the body waves.

Body waves in turn can be categorized into P-waves and S-waves.

3.1.4.1 P-waves

Primary waves or P-waves are longitudinal or compression waves. When a longitudinal wave is passing through a medium the temporary direction of movement of the particles inside the medium is the same or opposite compared to the wave propagation direction. The result is a consecutive sequence of compressions and expansions in the wave propagation direction. See figure 3.1. Primary waves have the fastest propagation velocity of all seismic wave types. As a result in the event of an earthquake P-waves are felt or measured first. P-waves can travel through any type of material. Gasses support compression waves (sound) therefore P-waves can also travel through the air in the form of sound.

3.1.4.2 S-waves

Secondary waves or S-waves are transverse waves. When a transverse wave is passing through a medium the temporary direction of movement of the particles inside the medium is perpendicular to the wave propagation direction. See figure 3.1 Gasses and fluids do not support S-waves. Secondary waves do not propagate as fast as primary waves but S-waves are faster compared to the surface waves. Therefore in case of an earthquake S-waves are felt or measured shortly after the arrival of the P-waves. The latency between P-waves and S-waves is depended on the distance between the hypocentrum and the point of measurement but is usually in the order of seconds up to several minutes. The typical propagation velocity of an S-wave varies between 4 to 5 kilometers per second in the earths crust up to 7 kilometers per second in the inner mantle.

There are two important types of surface waves namely Rayleigh waves and Love waves.

3.1.4.3 Rayleigh waves

Rayleigh waves are named after sir John William Strutt the third Baron of Rayleigh who matematically predicted the existence of this type of surface waves in 1885. A Rayleigh wave has a movement pattern that is very similar to how waves move on a lake or on the middle of an ocean. When a Rayleigh wave is moving along the surface of the medium, particles in the medium temporarily move in ellipse shaped orbitals. The ellipse shaped movement is almost entirely in a two dimensional plane. The normals to the two dimensional orbital planes are perpendicular to the wave propagation direction. See also figure3.1. Particles deeper in the medium move in smaller orbitals. The propagation velocity of Rayleigh waves is approximately 90% of the propagation velocity of secondary waves. But because Rayleigh waves can only travel along the surface the distance traveled by Rayleigh is larger compared to the distance traveled by body waves.

3.1.4.4 Love waves

The second important type of surface waves are Love waves, named after A.E.H. Love a British mathematician who created a mathematical model for this type of surface waves in 1911. When a Love wave is moving along the surface of the medium, particles in the medium temporarily move from left to right perpendicular to the wave propagation direction and parallel to the surface of the medium. The amplitude of particle motion often decreases rapidly with depth. See figure 3.1. The Propagation velocity of Love waves is often slightly faster compared to Rayleigh waves, but slower compared to body waves. Love waves typically carry a lot of energy. Often in the event of a large earthquake Love waves are the waves that are felt and cause most of the damage.

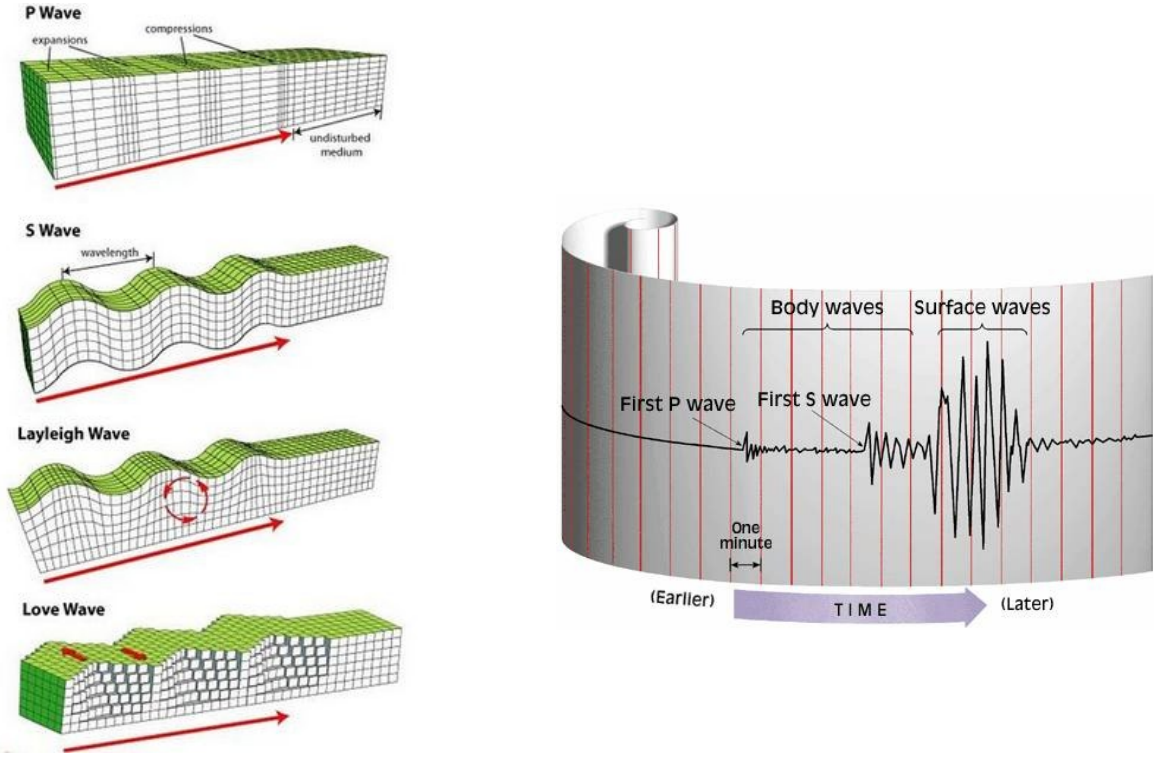

Figure 3.1 : Most common seismic wave types (left). A fictive (simplified) seismic recording (right).

3.2 Seismic wave medium

Seismic waves travel mostly through the earth. The earth can be though of as a medium for seismic waves. The earth is made of several layers. The outer most layer is the solid earth crust. The earth crust is part of the lithosphere. In general the lithosphere is the rocky and solid outer shell of a planet plus part of the uppermost mantle. The lithosphere is elastic when pressure is applied. But the lithosphere ruptures and is brittle when too much pressure is applied. Seismic waves are often caused by the rupture of the lithosphere. The rupture of the lithosphere can be thought of as one of several disturbances that cause seismic waves.

The layer directly beneath the earth crust is called the outer mantle or asthenosphere. The asthenosphere in turn is divided in two layers: The inner asthenosphere and the lowest part of the lithosphere. Unlike the lithosphere the inner asthenosphere is viscose and behaves like a very thick fluid. Although the inner asthenosphere is not solid it does allow S-wave propagation. The lowest part of the lithosphere is composed of the same material as the inner ashtenosphere but is much more rigid because of the lower temperature. The lithosphere “floats” on top of the asthenosphere. The temperature of the inner asthenosphere is estimated to be between 1400 and 3000 degrees Celsius. Subduction and plate tectonics in general are caused by convection currents in the inner asthenosphere and parts of the inner mantle.

The inner mantle is the layer directly beneath the outer mantle. The temperature of the inner mantle is estimated to be 3000 degrees Celsius. Although the temperature is higher compared to the inner asthenosphere most of the material in the inner mantle is solid. This is due to the immense pressure applied by the lithosphere and asthenosphere. Creeping slow viscous deformation of the material in the inner mantle is still possible.

The inner mantle encloses the outer core. Although the pressure on the outer core is even higher compared to the pressure on the inner mantle, the outer core is fluid. This is because the chemical composition of the material in the mantle is different from the chemical composition of the material in the outer core. Furthermore the temperature of the outer core is higher compared to the temperature of the inner mantle. The earths mantle is mainly composed of silicon oxide and magnesium oxide. The outer core is mainly composed of iron and nickel. The outer core temperature is estimated to range from 4400 degrees Celsius up to 5100 degrees Celsius. Being in a fluid state the outer core does not allow S-wave propagation.

The inner core of the earth is a primarily solid sphere composed of a nickel-iron alloy. The core is solid because of the gigantic pressure. Temperatures are estimated to vary between 5000 and 6000 degrees Celsius. These temperatures are similar to the temperatures at the surface of the sun.

The layered structure is mostly determined from studies of how seismic waves (S-waves and P-waves) behave as they pass through the earth.

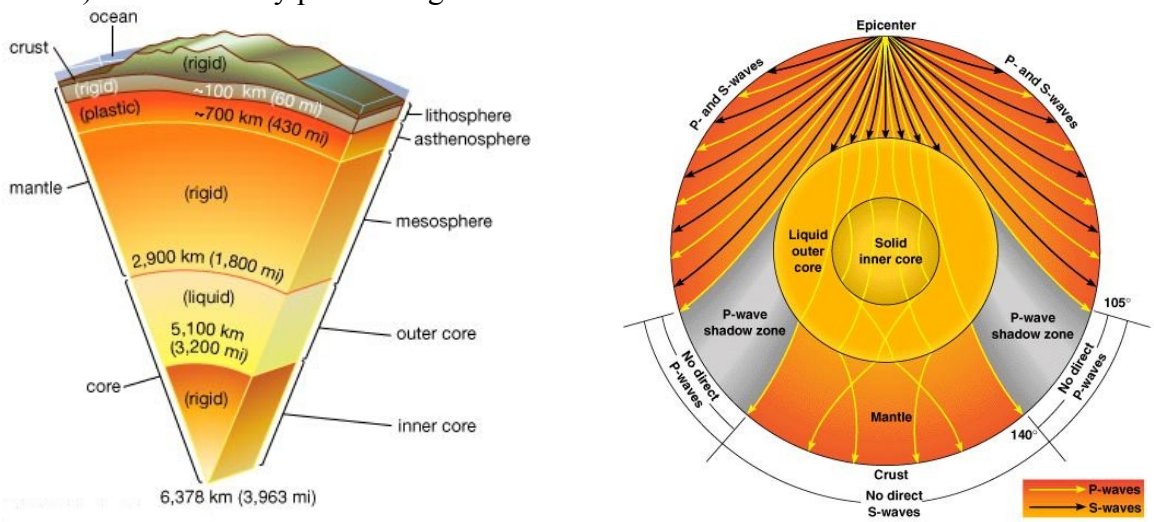

Figure 3.2: A schematic cut-out of the earth revealing the discussed layers (left). Propagation paths of P-waves and S-waves (right).

The propagation velocity of a seismic wave depends on the elasticity and density of the medium the wave is traveling through. The earth is composed of several layers with varying densities. As a result seismic waves travel with a varying propagation velocity between and within layers. For example the density of the inner mantle is higher close to the outer core compared to the density of the inner mantle close to the asthenosphere. In general the propagation velocity of seismic waves increases with depth whilst traveling through the same type of material (Pressure and density increases downward). See figure 3.2.

The path of a seismic wave within a layer is bend because of the gradient in density. At strong density edges between layers the path of a seismic wave is (almost) discontinuous. The path of a seismic wave is bend away from the normal of the density edge if the seismic wave is entering a denser material. Vice versa the path of a seismic wave is bend towards the normal of the density edge if the seismic wave is entering a less dense material. See also figure 3.2.

The two most important discontinuities are referred to as the Gutenberg-discontinuity and the Mohorovicic-discontinuity. The latter is also often abbreviated to Moho-discontinuity. The Moho-discontinuity is measured at a depth of between 20 to 40 kilometers below the earths surface for continental plates and 4 to 8 kilometers below the ocean floor. The Moho-discontinuity is believed to indicate the edge between the oceanic and continental crust and the underlying mantle. The Gutenberg-discontinuity occurs within the earths interior at a depth of approximately 2900

kilometers below the surface. At this depth there is again an abrupt discontinuity in seismic wave paths and velocities. It is believed that the Gutenberg-discontinuity indicates the edge between the solid inner mantle and the liquid outer core.

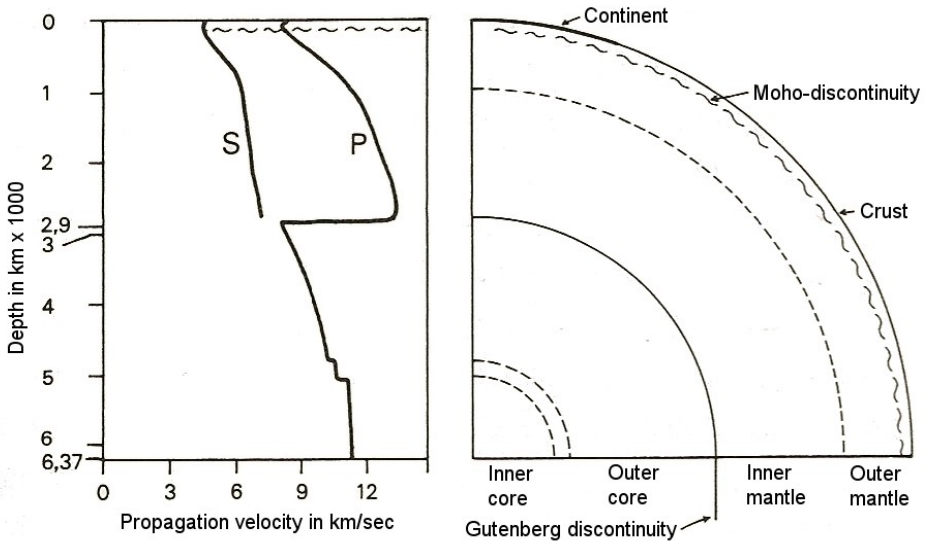

Figure 3.3: Body wave Propagation velocity versus depth. [2]

3.3 Seismic wave registration

Seismic waves often cause a temporary movement of the earths crust. The amplitude of this movement can be as much as several decimeters near the epicentrum. These kind of movements are certainly felt by humans. On larger distances from the epicentrum the amplitude of movement is often only a fraction of a millimeter. These kind of movements are certainly not felt by humans Accurate quantification of the amplitude of earth movement in time is very interesting in many fields of science. For example the knowledge about the layered structure of the earth is largely courtesy to accurate quantification of earth crust movements in time. Furthermore accurate quantification of earth crust movement also helped to locate sources of earthquakes such as subduction zones. In the field of volcanic seismology accurate seismic measurements help to predict future volcanic eruptions.

There are three main terms for seismic wave indication and registration devices.

The simplest of seismic wave registration devices are referred to as seismoscopes. Seismoscopes only register the occurrence of a seismic wave and perhaps provide the user with additional information such as a simple indication of the seismic wave magnitude. The first known seismoscope was invented in china by Zhang Heng in the year 123 AD. According to remaining texts the instrument was named “Houfeng Didong Yi“ which translates to instrument for measuring the seasonal winds and the movements of the earth. An ingenious mechanical system could release one of eight bales each suspended in the mouth of a bronze dragon to indicate the occurrence and direction of a passing seismic wave. Seismoscopes do not provide the user with a continuous recording of ground movement.

Seismographs and seismometers do provide the user with a continuous recording of ground movement in time. The terms seismograph and seismometer are often used to indicate the same type of instrument but the term seismograph is more applicable to the older type of instruments where

both the quantification and registration of earth movement is done in one instrument. Thus a seismograph is a instrument that translates and amplifies the ground movement often mechanically and also does the registration. Whereas the functions of quantification and registration are clearly separated in case of a seismometer.

The first difficulty in designing and constructing a seismograph or seismometer is to suspend part of the measurement instrument such that this part remains in a fixed position whilst the earth crust and everything that is attached to it moves. All seismographs and seismometers use a weight, often also referred to as the internal mass. The internal mass is somehow attached to the instruments frame and can move relative to it. In a very early (basic) seismograph design the internal mass is suspended from a tall fixture resulting in a pendulum. The pendulum can only move in its own resonance frequency which is dictated by the length of the pendulum. If the pendulum is long enough it will hardly move as a result of an earthquake. But the required length of the pendulum does not allow for compact instruments. Therefore in more complicated seismograph designs the internal mass is suspended using springs and or multiple anchor points to achieve the same low resonance frequency. In some modern seismometers the internal mass is suspended in a magnetic or electrostatic field. In these instruments the internal mass is kept nearly motionless (relative to the instruments frame) by a electronic negative feedback loop. The force required to keep the internal mass in place is a measure of earth crust movement.

The second difficulty in designing and constructing a seismograph or seismometer is to measure and register the movement of the instrument relative to the internal mass. In our very (basic) seismograph design the relative movement might be measured and registered using a pen that is attached to the internal weight and a paper transport mechanism attached to the instruments frame. See also figure 3.4 In a more involved seismograph design a registration device such as a pen might be attached to a set of levers that mechanically amplify the relative movement of the internal weight. Again the recording might be kept on a continuous sheet of paper that is fed through the instrument via a paper transport mechanism. Nowadays the movement of the instrument relative to the internal mass is measured electronically, the resulting electronic quantity is typically fed to an analog to digital converter (ADC) turning the analogous electronic quantity into a time discrete and amplitude discrete signal. The resulting digital seismic recording is typically stored in computer memory awaiting further processing and or inspection.

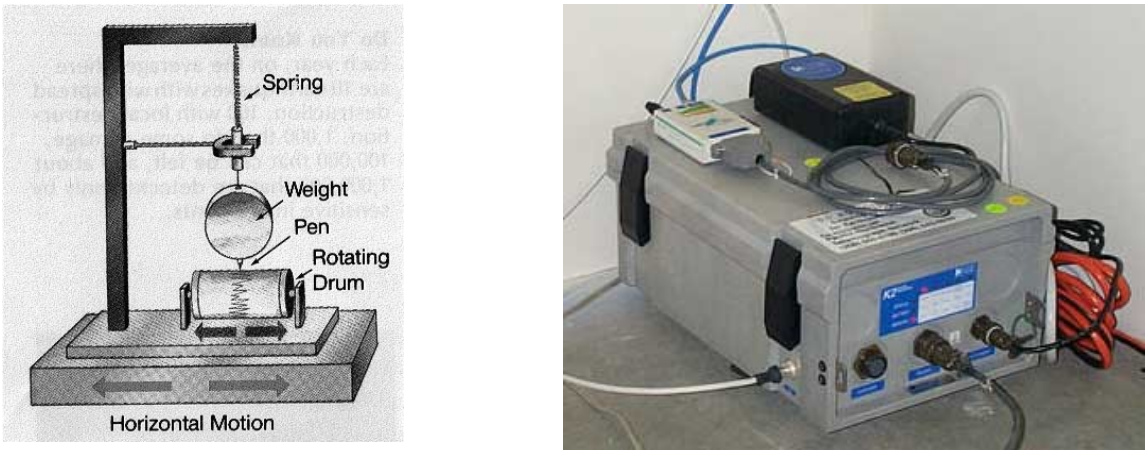

Figure 3.4: Basic seismograph design with pendulum (left). Modern seismometer with digital signal registration (right).

3.4 Volcanic seismic signals

The result of registering earth crust movement using a seismometer is called a seismic signal. Volcanic seismic signals are seismic signals that are caused by volcanic activity. Prior to an eruption a volcano typically produces many thousands of (small) earthquakes. Often earthquakes can be used successfully to predict inbound eruptions. Seismic signals that are specific to volcanic activity are caused by two main types of physical phenomena.

The withdrawal or injection of magma can cause pressure changes in solid rock. These pressure changes can cause the solid rock to break and crack. The resulting earthquakes are called volcanic tectonic earthquakes. Volcanic tectonic earthquakes are often an indication for renewed volcanic activity. Volcanoes can produce VT earthquakes for several days up to years prior to a possible eruption. Therefore volcanic tectonic earthquakes are not a reliable source for eruption prediction. Volcanic tectonic earthquakes or VT earthquakes often have a relative high frequency usually somewhere between one and five hertz.

The unsteady transport of magma through the cavities and folds in the earth can also cause a second type of earthquake. A sudden blockage in the path of the traveling magma can cause something that is similar to the “water hammer”. The “water hammer” can occur when water is traveling through a pipe and suddenly the passage is blocked for example by closing the tap. Instead of stopping instantaneously the water bounces against the tap and creates a pressure wave that moves back and forward through the pipe. Similar things can happen during the transport of magma inside a volcano. The resulting earthquakes are called long period earthquakes or LP earthquakes. Long period earthquakes often have a lower frequency compared to volcanic tectonic earthquakes usually between half a cycle per second up to three cycles a second. LP earthquakes are more informative to seismologists compared to VT earthquakes.

Volcanic tremor is a long period earthquake but one that lasts much longer than a long period earthquake. A single volcanic tremor can last from several minutes up to months. Frequency characteristics are similar compared to LP earthquakes.

Finally a hybrid earthquake is a combination of a volcanic tectonic earthquake followed by a long period earthquake or vice versa. The occurrence of a VT earthquake might trigger a LP earthquake or the other way around, the result is a hybrid earthquake.

Earthquakes that are caused by volcanic activity do travel the same way compared to normal earthquakes. Like normal earthquakes P-waves are the first to arrive at the point of measurement followed by S-waves and surface waves.

4 Dataset

4.1 Introduction

Prior to the eruption of the Nevado del Ruiz volcano in 1985 Colombian volcanoes were not daily and sufficiently monitored. As a response to the devastating eruption in 1985 the Colombian government decided to start monitoring potentially dangerous volcanoes more regularly. The institute responsible for monitoring these potentially dangerous volcanoes is the volcanological and seismological observatory at Manizales or VSOM abbreviated. Earth crust movement is measured and stored digitally from several strategic locations resulting in an increasingly large database of seismic recordings. Nowadays classification of seismic signals is done by the VSOM staff by visual inspection of the signals. Needless to say that this manual classification of regions of interest is time consuming and labor intensive.

Although the number of available recordings from the volcanological and seismological observatory is huge in this study a much smaller subset of segmented and labeled recordings is used because of practical and computational reasons. Thus the recordings in this smaller subset were first part of much longer continuous recordings possibly containing hours of less interesting background or noise signal. Furthermore a single label is assigned to each segmented recording. Both the segmentation and the labeling was done by human experts. The ground truth labels are used to test classification performance. The recordings in our subset are approximately one minute in length but recording lengths do vary. The recordings were digitized using a sampling frequency of 100.16 Hz and a amplitude resolution of 12 bits. The sampling values resulting from the 12 bit analog to digital converter are unsigned meaning that the sample values vary between 0 to 4095. The offset on the sample values should be approximately 2096 but often the offset deviates from this value.

In our subset recordings from five seismic stations are included. The seismic stations included are ALF, BIS, OLL, REC and REF. The recordings are coherent, meaning that any recording ^{i} corresponds to the same event for all seismic stations (Same event different measurement locations). The corresponding recordings are not necessarily exactly aligned in time though.

Four signal classes are included in our subset. The signal classes included are LP, RE,TL and VT. The VT and LP classes correspond to volcanic tectonic and long period seismic signals respectively. The cause of these signal classes was already discussed in the previous chapter (page 15). Because of the subduction process of the Nazca oceanic plate with the south American continental plate local tectonic earthquakes not caused by volcanic activity also do frequently occur. It is of interest to the VSOM staff to discriminate between local tectonic earthquakes not caused by volcanic activity and earthquakes that are caused by volcanic activity. Local tectonic earthquakes correspond to the TL (sismos Tectónicos Locales) signal class. The RE signal class correspond to regional seismic events. This class of seismic signals is very similar to the TL class. Regional seismic events also originate from an active fault. The difference between TL events and RE events is the distance between the hypocentrum and the point of measurement. RE events are more distant compared to TL events. A typical measure for hypocentrum distance is the difference between P-wave and S-wave arrival times.

The data-set received for this study also contains four additional signal classes but these signal classes are not included in this study because these signal classes are very poorly sampled in the received data-set. (poorly sampled = only a few examples per signal class)

In figure 4.1 a example of the time representation for each included signal class is given.

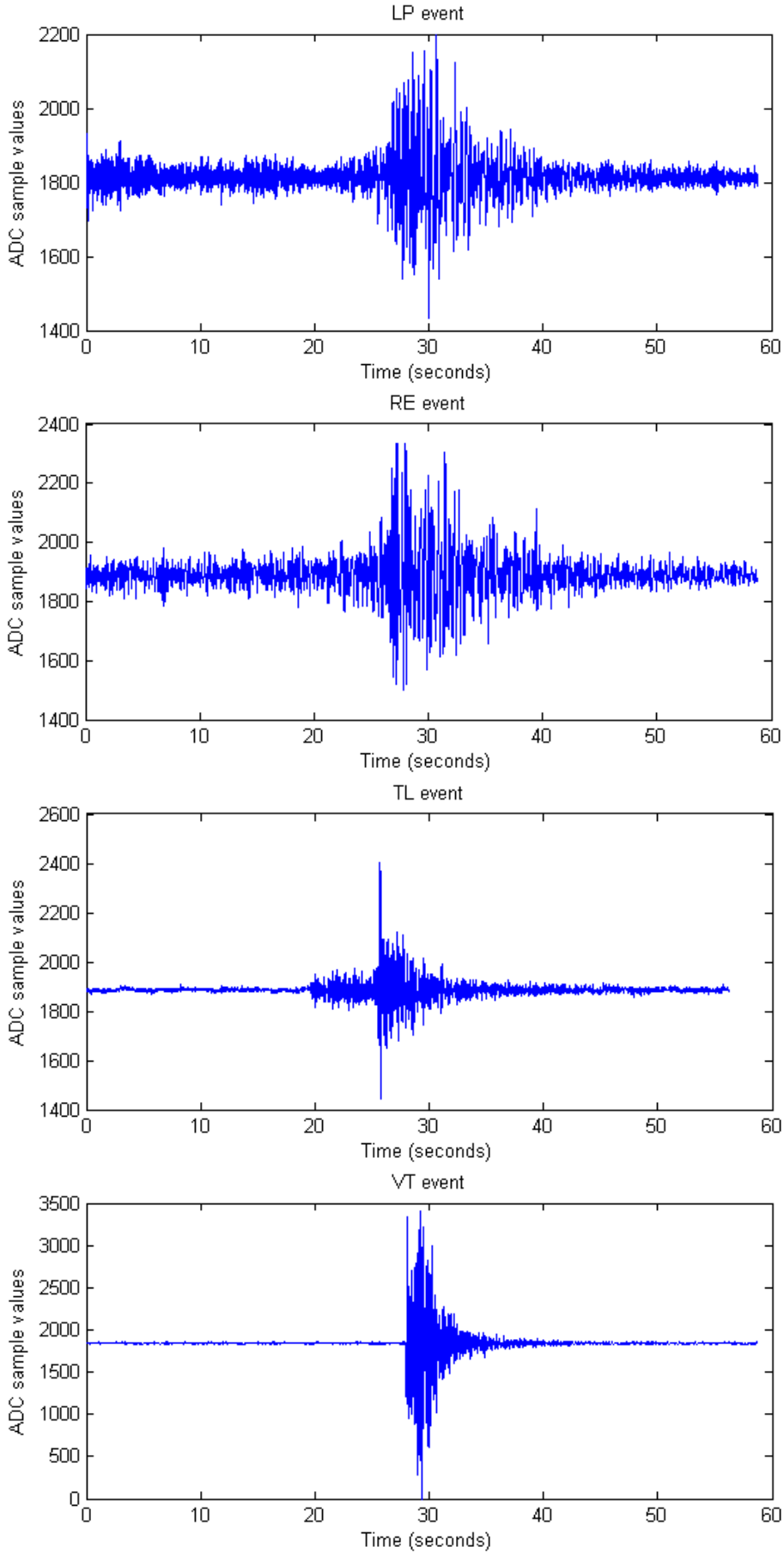

Figure 4.1: Example of a LP, RE,TL and VT event

5 Feature selection/extraction

5.1 Introduction

One of the fundamental questions concerning the representation of the seismic recordings is if it is better to use a single spectrum or to use a spectrogram per recording. In this chapter a justified dimensionality reduced spectrogram is developed using a set of existing signal processing and pattern recognition techniques. The resulting spectrogram representation is compared to the single spectrum representation using a Bayes classifier that does not assume/incorporate frame ordering.

In figure 5.1 a segmented seismic recording is produced by the sensor block. This recording is also referred to as the sensor representation. The dimensionality of the sensor representation is often too high for direct classification. A typical segmented seismic recording contains thousands of measurements or variables. A solution is to reduce the number of measurements.

Feature selection is concerned with selecting those d variables that contribute most to discrimination (d is an a priori chosen desired number of variables). An optimal feature selection solution is to evaluate all possible combinations of d variables using a chosen optimality criterion. The criterion function is over all possible combinations of d variables. But performing optimal feature selection directly on the sensor representation is often too computationally expensive even for a small number of variables. Sub optimal feature selection strategies exist. These sub optimal strategies reduce the number of evaluated combinations dramatically but are still impractical to apply directly on our sensor representation. Therefore in this study no feature selection techniques are used in the feature selection/extraction block. A combination of a feature extraction step followed by a feature selection step is possible. [12]

Feature extraction is a transformation of the sensor representation (using all variables) to a feature representation with a reduced number of variables. The criterion function is taken over all possible transformations of the variables. Of-course the number of possible transformations is very high possibly infinite. But usually the class of transformation is a priory specified, bounding the number of possible transformations. The result is that feature extraction techniques are often far less computationally expensive compared to feature selection techniques. Furthermore feature extraction techniques provide both linear and non-linear transformations of the sensor representation whereas feature selection techniques only provide axis aligned projections of the sensor representation. Therefore in this study feature extraction techniques are used to reduce the dimensionality of the sensor representation.

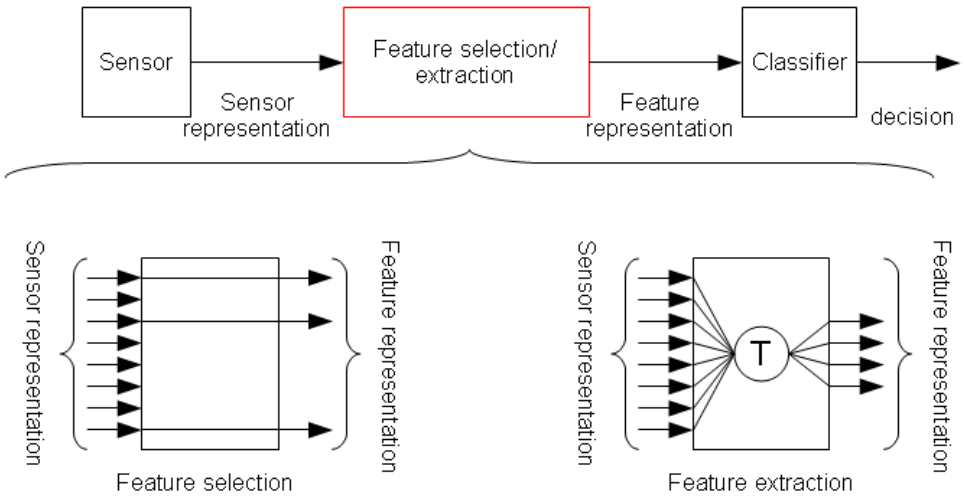

Figure 5.1: Feature selection and extraction and the pattern classifier

5.2 Feature extraction pipeline

To reduce the number of possible transformations in our feature extraction block a class of transformations is specified a priory. The class of transformations used in this study is a concatenation of (signal processing and pattern recognition) blocks. See also figure 5.2. Each block in the feature extraction pipeline modifies or transforms the sensor representation or intermediate representation (hopefully) towards a relevant set of features in the feature representation. A short explanation/justification for each block or pair of blocks is given below.

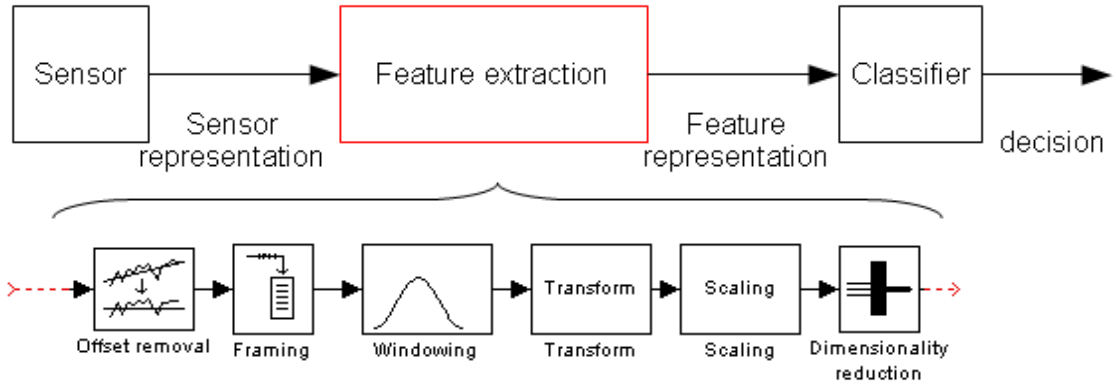

Figure 5.2: Feature extraction pipeline used in this study

5.2.1 Offset removal

The first block in our feature extraction pipeline is the offset removal block. The offset removal block is responsible for removing the sample offset that was introduced by the unsigned analog to digital converter. Some signal transformations assume a zero mean recording. In this study the sample offset is removed by calculating the mean value of the given segmented seismic recording followed by subtracting this mean value from the same seismic recording. One could also use a high pass filter to achieve approximately the same thing. But a typical high pass filter (with a flat magnitude response in the pass band) also introduces an undesired step response in each segmented seismic recording [7]. In case of a continuous recording the low pass filter would have been a better solution. The sample probability density functions (per class) of the mean values of our selected data-set are given in figure 5.3. Clearly the probabilistic distance between these classes is very small. This indicates that the mean values do not contribute to classification performance, and are therefore removed. A quantitative justification is given in the results section.

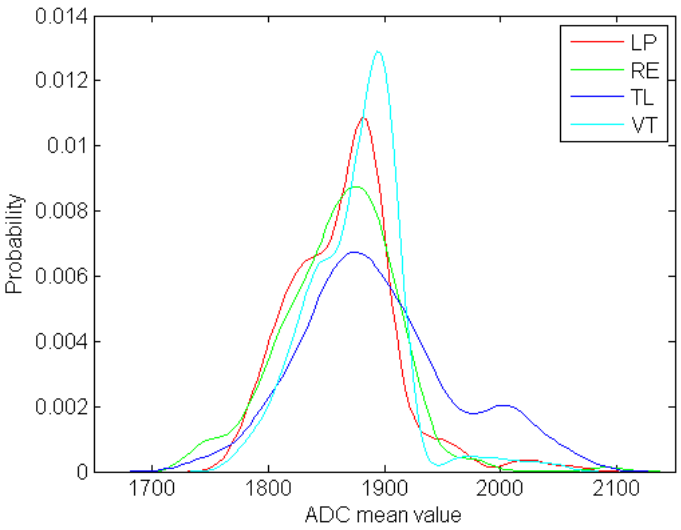

Figure 5.3: Sample probability density functions of the mean values

5.2.2 Framing

A typical segmented seismic recording contains several seismic wave types such as P-waves and S-waves. Therefore a typical segmented seismic recording is not stationary. Meaning that signal statistics such as the mean, variance and frequency information change over time. Therefore it is interesting to measure signal statistics in the segmented seismic recording at several instances in time. The process of dividing a longer time series such as our segmented seismic recording into shorter possibly overlapping frames is also referred to as framing. The frame length and the amount of overlap between consecutive frames are parameters of interest. Overlapping frames introduce more measurement and the amount of redundancy between measurements increases with the amount of overlap. The advantage of overlapping frames is an increase in time/transform space resolution, reducing the probability of missing short but important transients in the given recording.

5.2.3 Windowing and transform

In this study two frequency transformation methods are used.

5.2.3.1 Discrete Fourier transform

The first transformation method is a non-parametric discrete Fourier transform. The discrete Fourier transform or DFT abbreviated transforms a discrete and finite sequence (such as the frames resulting from the framing step) into its frequency representation by projecting this sequence on a finite set of discrete cosine and sine functions. The resulting frequency representation is complex. In this study the presented sequences are in time domain. The DFT assumes that the presented time domain sequences are exactly one period of a periodic signal and that this periodic signal repeats itself towards infinity in both directions. Thus the endpoints of the time domain signal are interpreted as if they were connected together. In a practical situation the time domain information in a given frame is seldom precisely one period of a periodic signal. The result is that the endpoints are discontinuous. The discontinuous endpoints introduce frequencies in the resulting spectrum that are not really present. This is also referred to as spectral leakage. Applying a window to the time domain information in each frame before computing the DFT can provide a better or smoother endpoint connection. The result is reduced spectral leakage. But spectral leakage reduction also reduces spectral resolution. Choosing a window function is always a trade-off between spectral leakage suppression and spectral resolution. The output of a DFT is half redundant when presented with a real input. Meaning that one can obtain complete information by only looking at approximately half of the complex outputs. The other half is removed. The magnitude squared is computed from the first half of the complex outputs. For computational reasons the fast Fourier transformation is used instead of the DFT. The FFT is more restrictive compared to the DFT but the resulting frequency representations are identical.

$$X[k]=\sum_{n=0}^{N-1} x[n]e^{-i(2\pi/N)kn} \qquad X[t,k]=\sum_{n=0}^{N-1} w[n]x[t+n]e^{-i(2\pi/N)kn}$$

Equation 5.1: Discrete Fourier transform for periodic signals according to [6][7] (left) Short time windowed Fourier transform (right)

5.2.3.2 Yule-Walker auto-regression

The second transformation method used in this study is a parametric auto-regressive method. Instead of projecting the given sequence on a finite set of discrete cosine and sine functions auto-regressive (AR) methods model the sequence as being the result of a linear model that is driven by white noise. The linear model is given in equation 5.2. The problem in AR analysis is to find good parameters (a_i) for the linear model given a sequence $x[n]$. Several methods of finding the parameters exist. In this study the AR method relies on the efficient inversion of the Toeplitz auto-correlation matrix using the Levinson-Durbin recursion. This AR method is also referred to as the Yule-Walker AR method. The Yule-Walker AR method assumes that the measurements outside the given finite sequence are zero. To avoid large prediction errors near the edges of the given sequence

again a window is applied to the given sequence before applying the Yule-Walker AR method. The AR method used in this study also assumes that the presented sequence is zero-mean. The resulting parameters can be used to estimate the magnitude spectrum of the given finite sequence. Or the parameters can be used directly for classification. The number of frequency bins in the magnitude spectrum can be arbitrarily high because the magnitude response is synthesized from the parameters. But in this study the number of frequency bins in the output is half the number of elements in the finite input sequence. (Equal number of frequency bins compared to the discrete Fourier transform) The number of parameters or filter coefficients used (model order) is a parameter of interest. A possible advantage of a AR spectrum estimation over the discrete Fourier transform is the possibility to control the complexity of the resulting magnitude spectrum using the model order. Furthermore the overall amplitude of the recording is expressed in one variable (e) .

$$x[n]=\sum_{i=1}^{P} a_i x[n-i]+e \qquad |H(e^{i\hat{w}})|=\left|\frac{\sqrt{e}}{1+a_1 e^{-i\hat{w}}+\cdots+a_p e^{-i\hat{w}p}}\right|$$

Equation 5.2: Linear autoregressive model (left). Magnitude spectrum estimation using filter coefficients a_i (right).

5.2.4 Scaling and dimensionality reduction

In this study three dimensionality reduction methods are used.

5.2.4.1 Discrete cosine transform

The first dimensionality reduction method used in this study is the discrete cosine transform or DCT abbreviated. The DCT is both data independent and unsupervised. The DCT is very similar to the discrete Fourier transform. But the resulting output of the DCT is real. The presented sequence is only projected (linearly) on a finite set of discrete cosine functions. Furthermore the discrete cosine transform assumes that the presented input is only one half of a periodic sequence. The other half is identical to the first half but mirrored. Thus one assumed period is the concatenation of the presented input directly followed by the presented input mirrored. The resulting assumed periodic signal does not have end point discontinuities and therefore no prior windowing operation is required. When the presented input is correlated the discrete cosine transform is able to retain a lot of (comparable to the data dependent principle component analysis) the original variance in a much smaller number of variables (dimensionality reduction) [8]. Usually most of the original variance is packed in the first couple of variables. A data independent selection of the DCT variables was used. The number of DCT variables used is a parameter of interest. Several variants of the DCT exist with slightly modified definitions. The DCT used in this study is given in equation 5.3. Prior to applying the DCT the presented sequence is scaled using a data independent logarithmic transformation.

$$X[k]=w[k]\sum_{n=0}^{N-1} x[n]\cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right) \qquad w[k]=\begin{cases}\sqrt{\frac{1}{N}} & k=0\\ \sqrt{\frac{2}{N}} & k=1\cdots N-1\end{cases}$$

Equation 5.3: Discrete cosine transform [8]

5.2.4.2 Principle component analysis

The second dimensionality reduction method used in this study is the principle component analysis. The principle component analysis or Karhunen-Loeve 1 transform is data dependent but unsupervised. Meaning that one does need to use a representative training set to find a good transformation. Furthermore a different training set will in general give you a different transformation. But no label information is used to find the transformation. The principle

component analysis is a linear projection of the given sequence $(\boldsymbol{x})$ on the sorted eigenvectors (A) of the sample covariance matrix $(\hat{\Sigma})$. See equation 5.4. The sorting order of the eigenvectors is determined by the magnitude of the corresponding eigenvalues. Larger eigenvalues are more important. There is no other linear transformation that can maintain more of the original variance in a small number of variables than the principle component analysis. The principle component analysis is sensitive to scaling. Therefore prior to applying the principle component analysis the input variables are scaled to zero mean and unit variance (Data dependent but unsupervised scaling). The number of principle components used is a parameter of interest.

$$\hat{\Sigma}=\frac{1}{n}\sum_{j=1}^{n}(\boldsymbol{x_j}-\boldsymbol{m})(\boldsymbol{x_j}-\boldsymbol{m})^T$$

$$eigenvectors(\hat{\Sigma})=[\boldsymbol{X_1},\boldsymbol{X_2},\cdots,\boldsymbol{X_k}]$$
$$eigenvalues(\hat{\Sigma})=[\lambda_1,\lambda_2,\cdots,\lambda_k]$$
$$sort(\{\lambda_1,\boldsymbol{X_1}\},\{\lambda_2,\boldsymbol{X_2}\},\cdots,\{\lambda_k,\boldsymbol{X_k}\})=[\boldsymbol{X_1},\cdots,\boldsymbol{X_k}]$$
$$\boldsymbol{A}=[\boldsymbol{X_1},\cdots,\boldsymbol{X_p}]\quad PCA=\boldsymbol{A}^T\boldsymbol{x}$$

Equation 5.4: Sample covariance matrix (left). Principle component analysis is the projection of the original sequence $(\boldsymbol{x})$ on the (p) most important eigenvectors (right).

5.2.4.3 Fisher mapping

The third dimensionality reduction method used in this study is the Fisher mapping. The Fisher mapping is also referred to as the Karhunen-Loeve 5 transform. The Fisher mapping is both data dependent and supervised. Thus a labeled training set is required to find a good transformation. The Fisher mapping is again a linear projection. The presented sequences are projected on the sorted eigenvectors of the matrix product of the inverse sample within class covariance matrix and the sample between class covariance matrix. See also equation 5.5. Again the sorting order of the eigenvectors is determined by the magnitude of the corresponding eigenvalues. Larger eigenvalues are more important. The within class covariance matrix is defined as the weighted sum of the sample class covariance matrices $(\hat{\Sigma}_i)$. The between class covariance matrix is defined as the weighted sum of squared class mean $(\boldsymbol{m_i})$ and sample mean $(\boldsymbol{m})$ difference matrices ($(\boldsymbol{m_i})$ and $(\boldsymbol{m})$ are both column vectors). The Fisher mapping maximizes (linear) class separability.

$$\boldsymbol{S}w=\sum_{i=1}^{C}\frac{n_i}{n}\hat{\Sigma}_i$$

$$\boldsymbol{S}b=\sum_{i=1}^{C}\frac{n_i}{n}(\boldsymbol{m_i}-\boldsymbol{m})(\boldsymbol{m_i}-\boldsymbol{m})^T$$

$$eigenvectors(\boldsymbol{S}w^{-1}\boldsymbol{S}b)=[\boldsymbol{X_1},\boldsymbol{X_2},\cdots,\boldsymbol{X_k}]$$
$$eigenvalues(\boldsymbol{S}w^{-1}\boldsymbol{S}b)=[\lambda_1,\lambda_2,\cdots,\lambda_k]$$
$$sort(\{\lambda_1,\boldsymbol{X_1}\},\cdots,\{\lambda_k,\boldsymbol{X_k}\})=[\boldsymbol{X_1},\cdots,\boldsymbol{X_k}]$$
$$\boldsymbol{A}=[\boldsymbol{X_1},\cdots,\boldsymbol{X_p}]\quad fisherm=\boldsymbol{A}^T\boldsymbol{x}$$

Equation 5.5: Sample within class covariance matrix and sample between class covariance matrix (left). Fisher mapping is the projection of the original sequence $(\boldsymbol{x})$ on the (p) most important eigenvectors (right).

In figure 5.4 The feature extraction pipeline is given with typical intermediate representations and feature representation. In figure 5.4 the uppermost subplot is the unprocessed sensor representation, and the subplot at the bottom of the figure is the resulting feature representation. In the third up to and including the sixth subplot intensity is color coded using a color map. Dark blue corresponds to small intensity values and dark red corresponds to large intensity values. (A colorbar was not included in this plot because of limited space)

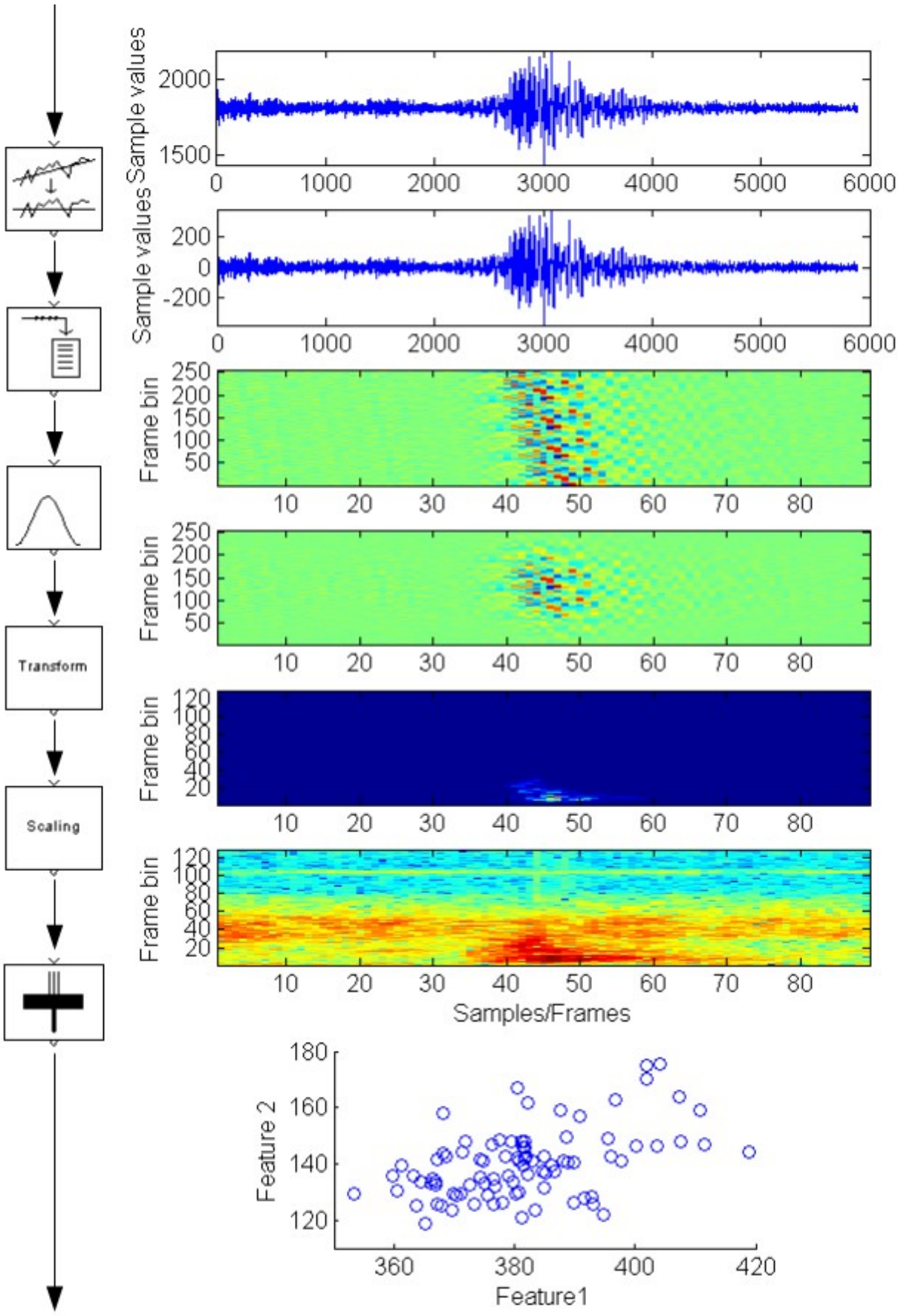

Figure 5.4: Feature extraction pipeline and typical intermediate representations and feature representation.

5.3 Experimental setup

5.3.1 Introduction

In the previous paragraph the class of transformations for the feature extraction block used in this study was discussed. But this does not already completely constraint the feature extraction pipeline. One can still choose among several blocks and combinations of blocks for the feature extraction pipeline. Furthermore the chosen blocks often also have parameters of interest that can vary. Two questions come to mind:

- What are good combinations of blocks?
- What are good parameters for the chosen blocks?

5.3.2 Testing criterion

One needs a criterion to decide which combinations of blocks and block parameters are good. Examples of these criteria are classification error/performance, class overlap/seperability and information loss.

Classification error/performance can be directly estimated using techniques such as the holdout estimate and Cross validation. The holdout estimate method splits the available data once into two mutually exclusive sets. These sets are often referred to as the training set and test set. The training set is used to train a classifier and if required to find a good transformation or projection for a data dependent dimensionality reduction method such as the PCA or Fisher mapping. The test set is only used to test classification error/performance. The cross validation method repetitively splits the available data into a mutually exclusive training set and test set. Each example in the available data is used only once for testing. Thus computational complexity increases with a decreasing test set size. A relatively large training set results in a well trained classifier but a unreliable error/performance measurement. Vice versa a relatively large test set results in a good error/performance measurement but this measurement is obtained using a possibly insufficiently trained classifier. Choosing a training set/test set proportion is always a tradeoff between the two. Often the available data is split in a 50% training set and 50% test set.

Probabilistic distance measures use the class conditional density functions to measure class overlap/separability. When the class conditional density functions can be estimated well, probabilistic distance measures are a very good indication for classification performance on unseen data. Furthermore probabilistic distance measures do not assume any type of distributions. One of the main disadvantages of the probabilistic distance measures is that they require an (accurate) estimate of the class conditional density functions. Second they also involve numerical integration which can be very very expensive if no explicit class conditional density functions are available. Most of the probabilistic distance measures simplify when a normal distribution is assumed. But this assumption can not always be justified.

Finally several easy to compute scatter based criteria exist. These criteria are based upon the sample within class covariance matrix, the sample between class covariance matrix and the (unsupervised) covariance matrix. When using scatter based criteria often the aim is to find as set of features for which the within class spread is as small as possible and the between class spread is as large as possible. The scatter based criteria often express the within-class spread and the between class spread in one single number.

In this study the criterion used is classification performance. The best classification performance estimates are the direct estimates such as Cross validation and the holdout estimate. These methods are far more expensive compared to the easy to compute scatter based criteria but provide

better classification performance estimates. In this study the probabilistic distance measures are not practical because the explicit class conditional density functions are unknown.

In this study the holdout estimate was used. The holdout estimate was repeated ten times each with a different random permutation of the data. The resulting ten classification performances were averaged to reduce the performance estimate variance and bias. The Mersenne twister random number generator was used in creating the random permutations of the data. In this study prior to each experiment the random number generator was seeded with a fixed seed. Thus all the permutation sequences were identical for each experiment.

5.3.3 Data set

In all the experiments the selected data were split in 75% for training and 25% for testing. From the received data set the first 133 examples or events were selected from each class. Thus 100 examples per class were used to train the classifier and if applicable to find a good data dependent transformation. The other 33 examples were only used for testing. Furthermore for these experiments only the examples originating from the OLL station were used. For these experiments equal class priors were assumed. This assumption is not supported by the empirical class frequencies in the received data set. In the received data set volcanic tectonic (VT) events, long period (LP) events and local tectonic events (TL) have approximately equal class frequencies. But in the received data set regional events (RE) occur approximately three times as often.

5.3.3 Classifier

Because the holdout method was used to estimate classification performance, one also requires a classifier. For these experiments the Bayes classifier was used. When presented with the true class posterior probability density function the Bayes classifier is optimal. Or in other words one can not attain better classification performance when the true class posterior probability density function is known. The Bayes classifier is a simple and very flexible classifier often providing very good classification performance, especially when there are a lot of training examples in a low dimensional feature space (Which is the case in these experiments).

In practice the true class posterior probability density function is not known. Instead a multivariate nonparametric kernel density method was used to estimate the class conditional density functions from the training set (see also equation 5.6). One class conditional density function was estimated per class. Thus for each random permutation of the selected data four class conditional density functions were estimated. A nonparametric kernel density method was used because visual inspection of the selected examples in their resulting feature space did at least not always suggest towards a parametric density function. When using a kernel density method one also needs to decide on the kernel to use. In these experiments a multivariate normal kernel was used. In practice the normal kernel is used most often. The choice of the kernel is not critical but the normal kernel might give slightly better performance because of its infinite extend in the feature space. Additional computational requirements of the normal kernel are not so much of an issue anymore.

The smoothing parameter was optimized for each presented training set (one smoothing parameter per class) using likelihood cross validation on the training set [3]. Likelihood cross validation was repeated ten times and the resulting smoothing parameters were averaged. Likelihood cross validation was used because it does not assume an underlying distribution. A bounded interval greedy search algorithm was used to maximize the likelihood as a function of the smoothing parameter during likelihood cross validation. Although there is no guarantee that the likelihood function only has one global maximum, inspection of the likelihood function on several random permutations always showed a well behaved function with one global maximum and no local maxima (see also figure 5.5). One smoothing parameter for all dimensions was used because of computational reasons. Optimizing a high number of smoothing parameters using likelihood cross validation is very time consuming.

To reduce the time required for these experiments a highly optimized GPU (graphics processor unit)

implantation was developed and used for these experiments. Kernel density estimation is massively parallel and scales extremely well on GPU architectures[4]. The typical speedup over my optimized single threaded c implementation was approximately 200x (Intel core 9550 vs nVidia 8800 GTS). Note that the GPU implementation uses single precision floating point math and fast approximations for the exp and log functions. But the differences between the resulting densities was marginal.

All frames within a single example or event were assumed to come from the same distribution. Furthermore all the frames within a single example or event were assumed to be independent of each other. Thus frame probabilities were computed independent of each other using one conditional density function per class. The probability on a single example or event is the product off the assumed independent frame probabilities. Because of numerical precision log probabilities were used. This approach allows for segmented seismic recordings of variable length. In figure 5.5 an example of a typical class conditional density function for a given training set (blue circles) is given. Each blue circle in figure 5.5 is a single frame from one of the training examples. The number of frames varies per training example. Furthermore the number of frames also depends on the window length and the amount of overlap between frames. One single training example is highlighted in green. The estimated multivariate probability density function is color coded. Dark blue corresponds to small log probabilities and dark red corresponds to large log probabilities.

$$p(\boldsymbol{x}|\omega_j)=\frac{1}{nh^p}\sum_{i=1}^{n}K\left(\frac{1}{h}(\boldsymbol{x}-\boldsymbol{x}_i)\right)\quad \boldsymbol{x}_i\in\omega_j \qquad K(z)=\frac{1}{(2\pi)^{p/2}}\exp\left\{-\frac{z^T z}{2}\right\}$$

Equation 5.6: Multivariate nonparametric kernel density estimation formula (left). Multivariate normal kernel (right).

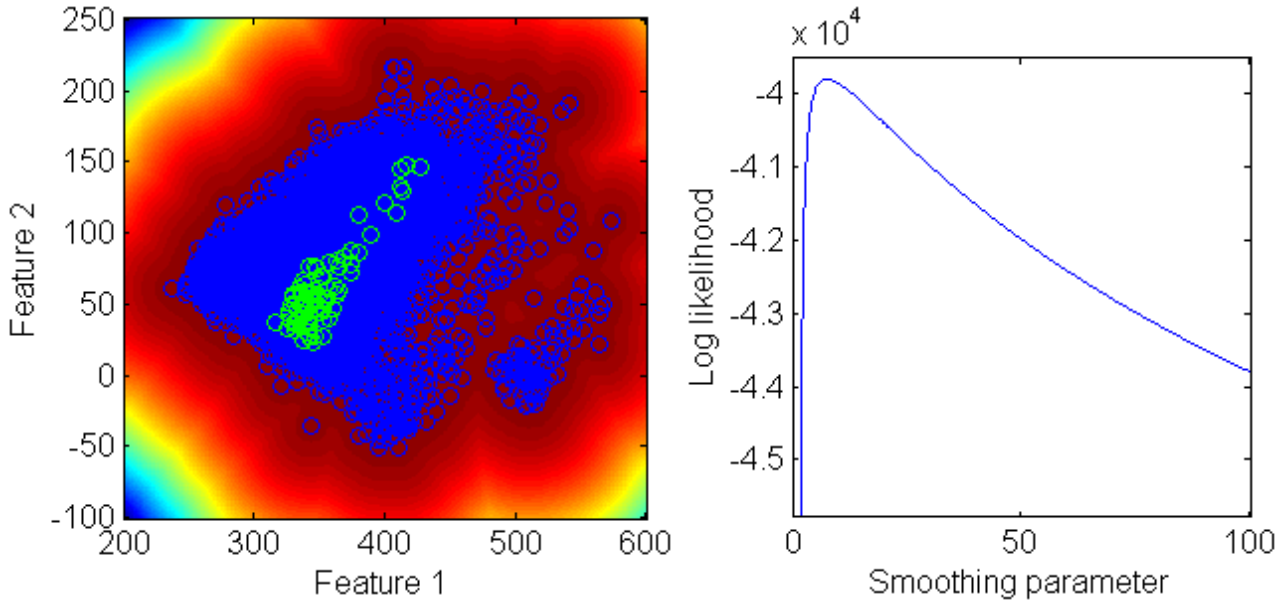

Figure 5.5: Two dimensional class conditional density estimation on a given training set (left). log likelihood as a function of the smoothing parameter showing a well behaved function with one global optimum (right).

5.3.4 Accuracy of performance estimates

Although pessimistically biased, using the hold-out estimate one can easily compute a confidence interval on the true classification performance using a set of independent test samples drawn from the same distribution as the training set. According to [3] the conditional density function of the true classification performance (cp_T) is binomial distributed (see equation 5.7). Actually in [3] the class conditional density function is given for the true error rate but both functions are identical. Using the Bayes rule and the assumption that the true classification performance does not depend on the

number of test examples one can find the posterior density function on the true classification performance (see also equation 5.7). In equation 5.7 n is the number of test samples used to estimate the true classification performance. k is the number of correct classified objects. In these experiments the number of objects used to estimate classification performance is 132. Using the posterior density function from equation 5.7 one can find a maximum 0.95 confidence interval of length 0.168 for k equals 66. This is the largest confidence interval for a single holdout estimate using 132 test samples.

$$P(k|cp_T,n)=\binom{n}{k}cp_T^k(1-cp_T)^{n-k} \qquad P(cp_T|k,n)=\left(\frac{cp_T^k(1-cp_T)^{n-k}}{\int cp_T^k(1-cp_T)^{n-k}\,dcp_T}\right)$$

Equation 5.7: Conditional density function of the true classification performance (left). Posterior density function of the true classification performance (right).

However, in these experiments the holdout estimate was repeated ten times. The ten resulting classification performances were averaged reducing the bias and standard deviation of the performance estimates. Several statistical summary tests such as the Shapiro-Wilkinson tests show that a typical sample of ten classification performances are normally distributed. Therefore one can use the student distribution to compute confidence intervals for the mean (the variance is unknown) The confidence interval is given in equation 5.8. In equation 5.8 $\bar{x}_n$ Is the sample mean and s_N is the sample standard deviation.

$$\left(\bar{x}_n-t_{n-1,\alpha/2}\frac{s_n}{\sqrt{N}},\bar{x}_n+t_{n-1,\alpha/2}\frac{s_n}{\sqrt{N}}\right)\quad\left(\bar{x}_n-2.228\frac{s_n}{\sqrt{10}},\bar{x}_n+2.228\frac{s_n}{\sqrt{10}}\right)$$

Equation 5.8: Confidence interval for a normal distribution with unknown variance (left). 0.95 Confidence interval for a normal distribution with unknown variance using ten samples (right).

A typical example of classification performance estimates as a function of a parameter of interest are given in figure 5.6. In figure 5.6 the upper and lower boundaries of the confidence interval are given (dotted red lines) Clearly in this example classification performance does not depend much on this chosen parameter of interest. But classification performance does seem to improve slightly towards the higher numbers for the parameter of interest. In these experiments the typical 0.95 confidence interval length equals 0.05 or 5%.

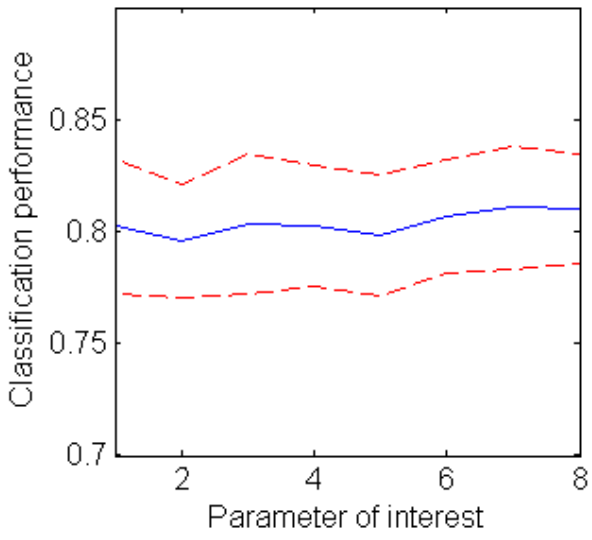

Figure 5.6: Typical example of classification performance estimates as a function of a parameter of interest.

5.4 Results

5.4.1 Magnitude squared FFT and DCT

In figure 5.7 experimental results of the block combination magnitude squared FFT and DCT are given as a function of the window overlap and window length (all other parameters remained fixed). A Hamming window was used for all the experiments. Furthermore the spectra were projected on the first four DCT components. The window overlap is given on the horizontal axis and the window length is given on the vertical axis. Classification performance is both color coded and given quantitatively. A window overlap of 0 means that there is no overlap between consecutive frames but there are also no unused samples between two neighboring frames. The amount of window overlap is proportional. For example a window overlap of 7/8 in combination with a window length of 256 means that consecutive frames are overlapping with 224 samples. In figure 5.7 classification performances are close thus for this block combination the window overlap and window length are not really critical. Or at least not as critical as one might expect. However classification performance is marginally better for a window overlap of 3/4 in combination with a window length of 256 and a window overlap of 7/8 in combination with a window length of 128. Clearly the data dependent and unsupervised feature extraction method performs best with a relatively high number of resulting frames in the feature space. But a window length of 64 is probably too short to capture important signal characteristics. This window length is therefore not included in the other experiments (this window length performed worst and also required most computations).

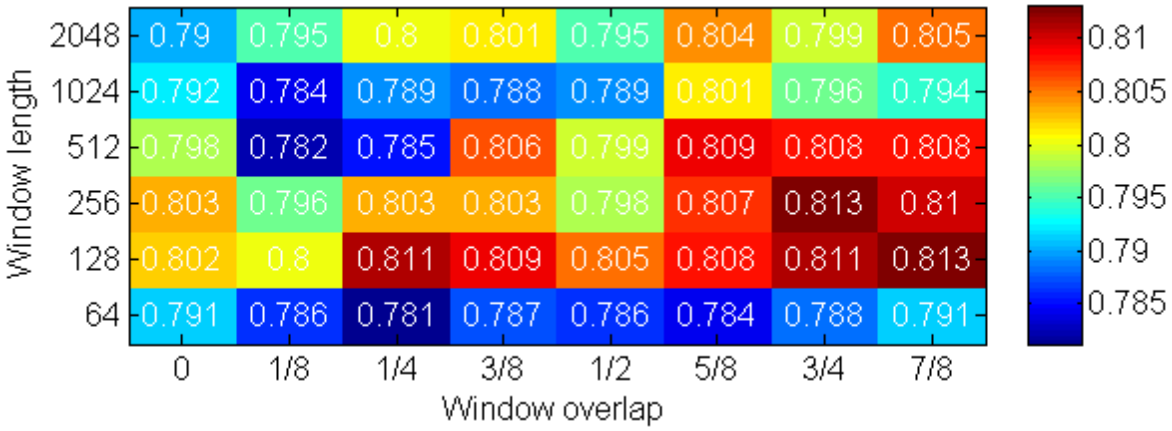

Figure 5.7: Classification performance as a function of the window length and window overlap using a discrete Fourier transform in combination with the DCT.

5.4.2 Magnitude squared FFT and PCA

In figure 5.8 experimental results of the block combination magnitude squared FFT and PCA are given as a function of the window overlap and window length. Again classification performances are close. The resulting spectra were projected on the first four principle components. Using more principle components did not improve performance. Again a Hamming window was used. In figure 5.8 a completely different pattern emerges. The data dependent but unsupervised feature extraction method clearly performs better with longer window lengths. This is also what one would expect. A high number of frames could introduce significant but undesired within class projection directions. Furthermore it might be possible that longer window lengths result in slightly more discriminative feature vectors compared to the shorter window lengths. And that the supervised feature extraction method is able to take advantage of these more discriminative feature vectors whereas the unsupervised method can not. Best classification performances using this method were obtained using a window overlap of 1/2 and a window length of 2048.

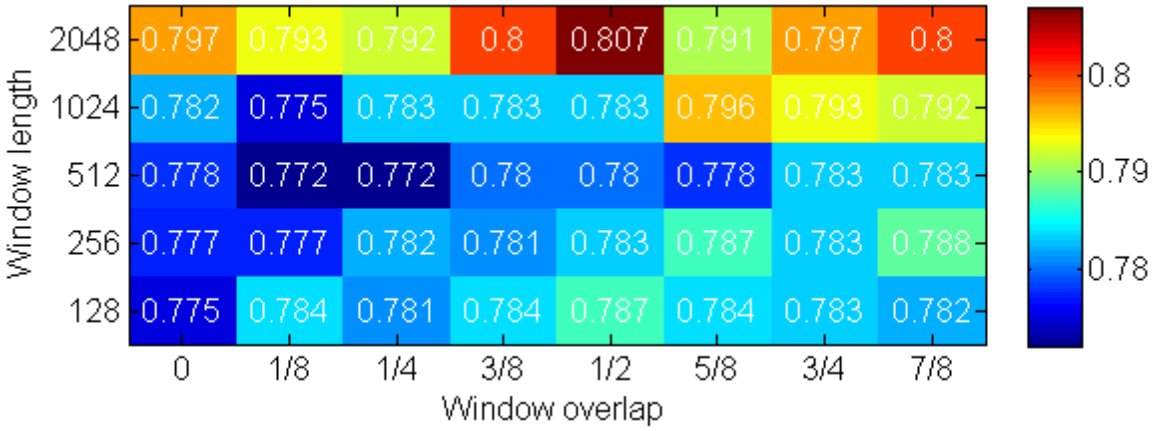

Figure 5.8: Classification performance as a function of the window length and window overlap using a discrete Fourier transform in combination with principal component analysis.

5.4.3 Magnitude squared FFT and Fisher mapping

In figure 5.9 experimental results of the block combination magnitude squared FFT and Fisher mapping are given as a function of the window overlap and window length. Because a Fisher mapping is used the resulting magnitude spectra were projected on a three dimensional feature space (largest possible number of dimensions for a four class problem). Most of the classification performances are close but for very long window lengths in combination with no or little window overlap classification performances are very poor. This is because of poor within class covariance matrix estimation. For a window length of 2048 in combination with a window overlap of 0 a typical segmented seismic recording is only partitioned into three frames. Best classification results were achieved using a long window length in combination with a large window overlap.

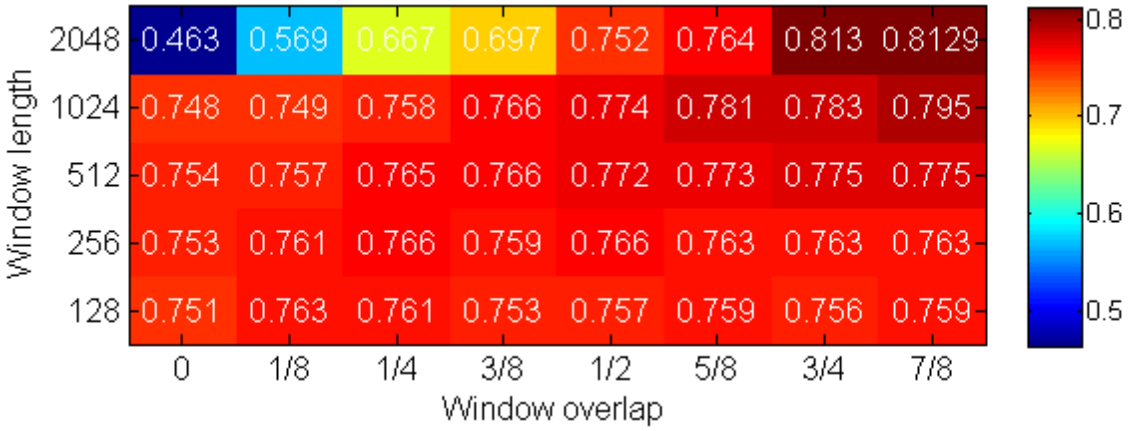

Figure 5.9: Classification performance as a function of the window length and window overlap using a discrete Fourier transform in combination with the Fisher mapping.

5.4.4 Yule-Walker auto-regressive model

In contrast to the FFT, using an auto-regressive method to estimate the power spectrum of a finite sequence one can control the complexity of the resulting spectra using one parameter (the model order). A small model order results in smooth but less detailed spectra. Whereas a large model order results in detailed spectra. Typically the averaged residual prediction error is used in combination with a simple heuristic to determine a suitable model order [13]. A typical averaged residual

predication error curve is given in figure 5.10. This error curve is the result of applying the Yule -Walker auto regressive method for several model orders on a typical random permutation of the selected data. A good choice for the model order according to the residual prediction error curve might be just to the right of the bending point of the curve e.g. a model order of 40 (After this point residual prediction error does not improve much). In figure 5.10 classification performance is also given as a function of the model order. According to figure 5.10 classification performance is maximized for a model order of 110. Classification performance was measured using the Yule-Walker auto regressive method in combination with the DCT. The window length was 256 and the window overlap was 0.75. Again a Hamming window was used. Of-course the optimal model order is dependent on the other parameters such as the window length and window overlap. The experiment was repeated once with a window length of 2048 and a window overlap of 0.75. In this experiment relatively good performance was achieved using a model order of 1050 and higher. Both results suggest that a good model order is approximately half the window length (Finding the optimal model order for each set of parameters is computationally too expensive). But from these results one can already conclude that detail in the resulting spectra is important for classification performance.

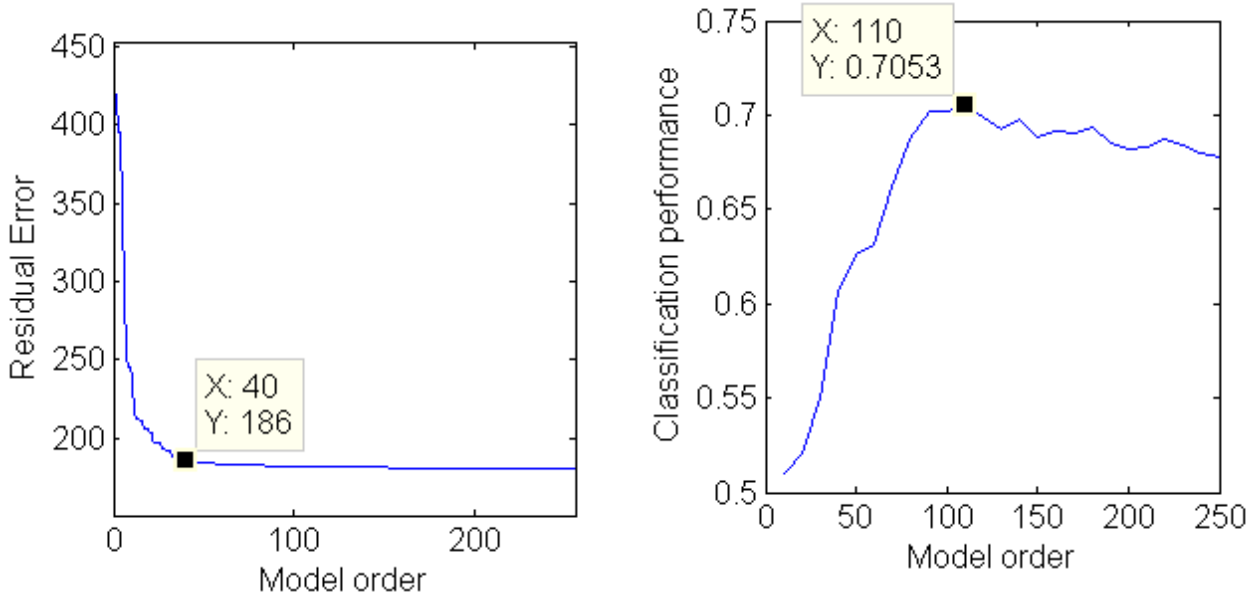

Figure 5.10: Residual prediction error as a function of the model order (left). Classification performance as a function of the model order (right).

5.4.5 Yule-Walker auto-regressive model and DCT

In figure 5.11 experimental results of the block combination Yule-Walker auto-regressive model and DCT are given as a function of the window overlap and window length. The model order for the Yule-Walker auto regressive block was set to half the window length for each experiment. Again comparable to the FFT DCT block combination best classification performance was achieved using short window lengths. But in this experiment classification performance was much worse for longer window lengths. The window length of 64 performed slightly better. Overall classification performances were not as good compared to the FFT DCT block combination.

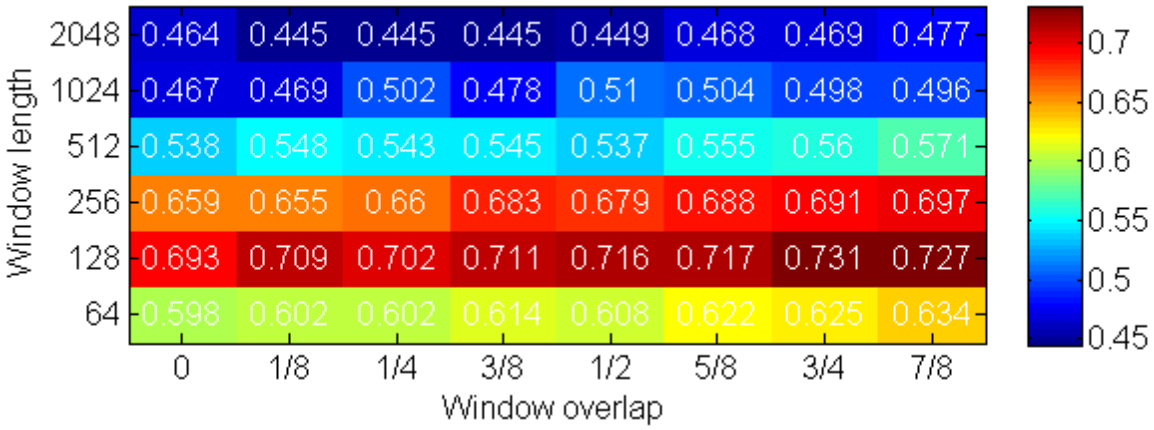

Figure 5.11: Classification performance as a function of the window length and window overlap using the Yule-Walker auto regressive method in combination with the DCT.

5.4.6 Yule-Walker auto-regressive model and PCA

In figure 5.12 experimental results of the block combination Yule-Walker auto-regressive model and PCA are given as a function of the window overlap and window length. The emerging pattern of classification performances is very similar to the FFT PCA combination. Overall classification performances are slightly better.

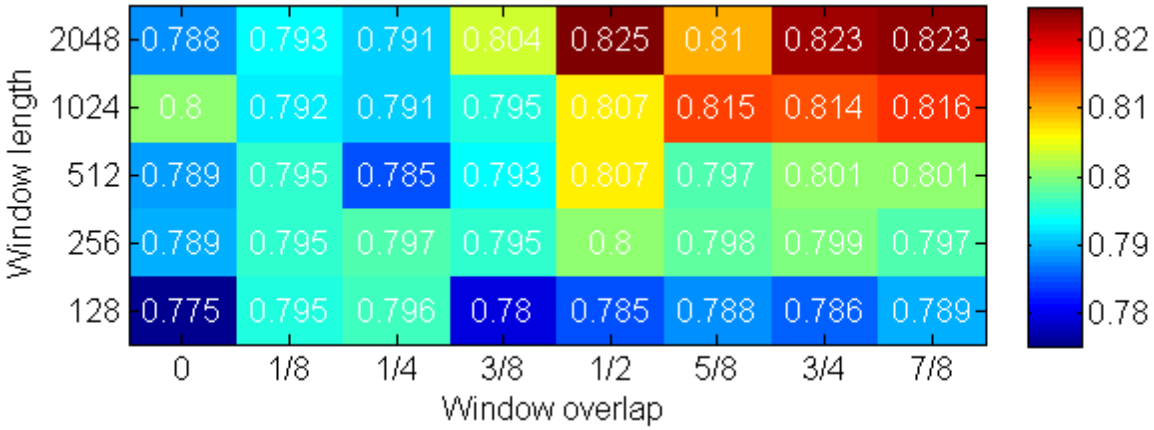

Figure 5.12: Classification performance as a function of the window length and window overlap using the Yule-Walker auto regressive method in combination with the data dependent but unsupervised PCA.

5.4.7 Yule-Walker auto-regressive model and Fisher mapping

In figure 5.13 experimental results of the block combination Yule-Walker auto-regressive model and Fisher mapping are given as a function of the window overlap and window length. Overall classification performances are slightly better for the Yule-Walker auto-regressive method in combination with the Fisher mapping compared to the FFT Fisher mapping combination. But other than that results are the same.

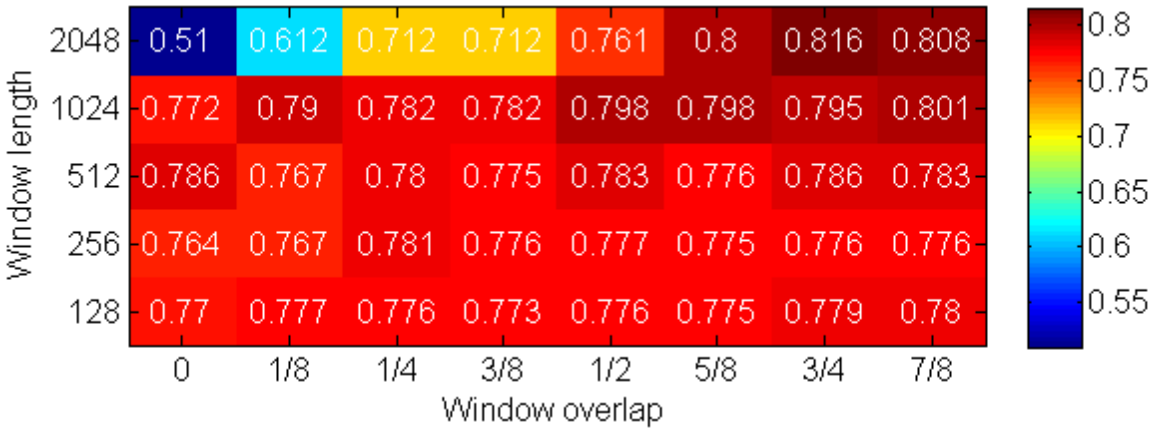

Figure 5.13: Classification performance as a function of the window length and window overlap using the Yule-Walker auto regressive method in combination with the data dependent and supervised Fisher mapping.

5.4.8 Influence of the window type on classification performance using the magnitude FFT in combination with the DCT

In the previous paragraphs test results for several block and block parameters were given. Best classification performance was achieved using the Yule-Walker auto regressive method in combination with the data dependent PCA. But these results were obtained after the discovery that the residual prediction error is at least not always a good indication for retained classification performance. Prior to this discovery best classification performance was achieved using the magnitude squared FFT in combination with the data independent DCT. All of the coming tests were already performed prior to this discovery.

Thus for the coming tests the magnitude squared FFT was used in combination with the DCT despite the fact that better classification performances were achieved using the LPC in combination with the PCA. A window length of 256 and a window overlap of 0.75 was used.

In figure 5.14 Classification performance for this block and parameter combination is given using several different window types. In figure 5.14 The window type is given on the horizontal axis on the vertical axis classification performance is given. Again classification performance is close. Best classification performance was achieved using a Hamming window. This does not come completely as a surprise because both the block and parameter combinations were optimized using this window type. Because of computational constraints one cannot try all combinations of blocks and block parameters. If one would have performed the optimization tests using a different window type probably classification performance would have been better or even best for this different window type. Worst classification performance was achieved using a rectangular window. This is equivalent to not using a window at all. Classification performance of the rectangular window type was 0.721.

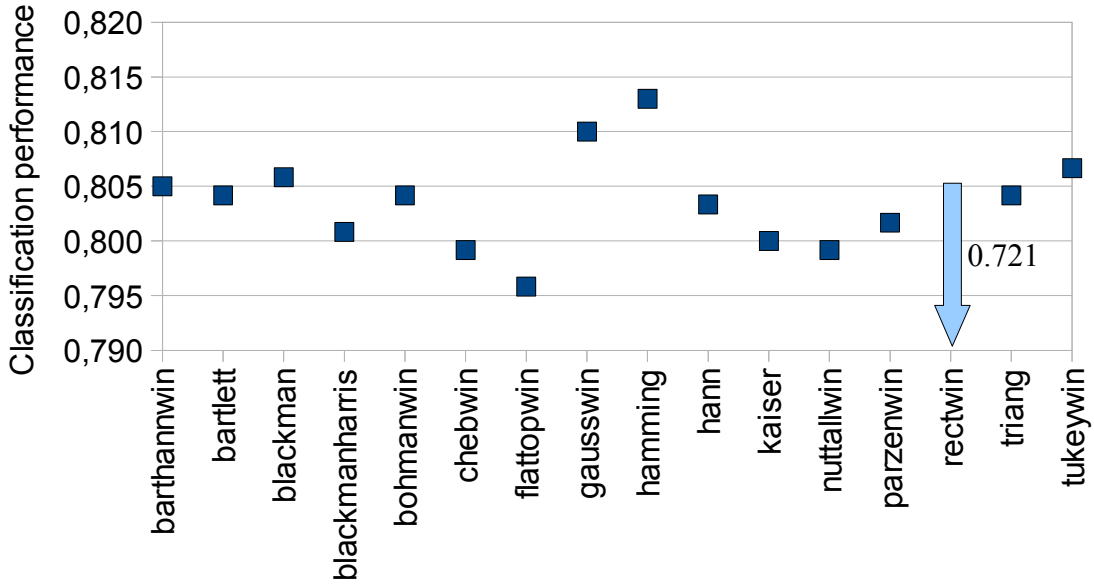

Figure 5.14: Classification performance as a function of the window type

5.4.9 Influence of the feature dimensionality on classification performance using the magnitude FFT in combination with the DCT

In figure 5.15 Classification performance is given for the magnitude FFT and DCT block combination as a function of the feature dimensionality. Best classification performance was achieved using a DCT feature dimensionality of four. But again classification performances were very close (for dimensionalities of four and more) . Similar to the window type experiment, block combinations and parameters were optimized using a DCT feature dimensionality of four. This might explain the marginal optimum for this feature dimensionality. Classification performance decreases for (very) high DCT feature dimensionalities.

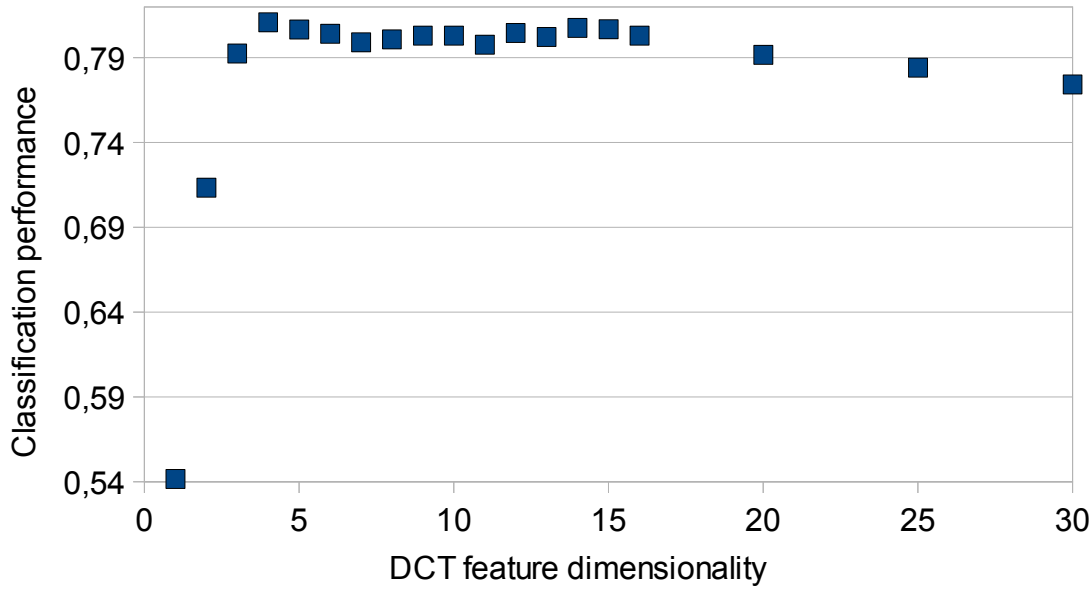

Figure 5.15: Classification performance as a function of the DCT feature dimensionality

5.4.10 Influence of the offset removal on classification performance

Removing the offset does improve classification performance a little when using the magnitude FFT in combination with the DCT (see table 5.1). The classification improvement is caused by the removal of variance that does not contribute to classification performance (see also figure 5.3 on page 18). Classification improvements are much greater for the Yule-Walker auto-regressive method (see also table 5.1). With the offset present predication errors are larger near the edges of the frames when using the Yule-Walker AR method, resulting in worse classification performance.

	FFT+DCT	Yule Walker AR+DCT
with offset	0,808	0,617
without offset	0,813	0,691

Table 5.1: Influence of the offset removal on classification performance

5.4.11 Classification performance using a single averaged spectrum per segmented seismic recording

One of the question of interest is, if it is required to divide the given segmented seismic recordings into shorter possibly overlapping frames or that a single averaged spectrum is already sufficient.

In this experiment the magnitude FFT was used in combination with the DCT. Per segmented seismic recording several frames are computed just like in the other experiments but in this experiment all frames resulting from one recording are averaged. Resulting in one averaged spectrum per recording instead of a spectrogram per recording. The window overlap used in this experiment was one minus the window length. Resulting in a high number of frames in case the window length is shorter compared to the recording length. Test results for this experiment are given in table 5.2. Similar classification results were achieved compared to the previous experiments using a spectrogram.

Window length				
128	**256**	**512**	**1024**	**2048**
0,81	0,81	0,81	0,82	0,82

Table 5.2: Classification performance using a single averaged spectrum per segmented seismic recording.

5.5 Observations and Conclusions

From the results one can conclude and observe the following:

- Dividing a given segmented seismic recording into several shorter possibly overlapping frames does not improve classification performance over a single averaged spectrum per seismic recording when used in combination with the kernel density Bayes classifier that does not assume frame ordering. When dividing a segmented seismic recording into several shorter possibly overlapping frames one has both time and frequency information. When using one averaged spectrum per segmented recording one only has frequency information. In both cases all information from each recording was used.
- Classification performance in most cases is at least not strongly influenced by the chosen window overlap and window length (except for the Yule-Walker AR DCT combination). Typically classification performance varies 3%-5% as a function of the window overlap and window length.
- The data independent and unsupervised dimensionality reduction method performed well in comparison to the data dependent PCA and supervised Fisher mapping. Best classification performance was approximately equal for the three dimensionality reduction methods. (Again except for the Yule-Walker AR DCT combination)
- The DCT performed best with a relatively short window length in combination with a large window overlap.
- Both the PCA and the Fisher mapping performed best with longer window lengths. The fisher mapping performed best with a long window length in combination with a large window overlap.
- Classification performance is not strongly influenced by the choice of the window type. The window type that performed best was also the window type that was used to optimize the other block parameters and block combinations. Using a window does improve classification performance significantly.
- Removing the mean from the segmented seismic recordings does slightly improve classification performance when using the magnitude squared FFT in combination with the DCT. The classification performance difference is greater for the Yule-Walker AR and DCT combination.

For this study several experiments were performed using the continuous wavelet transform. The continuous wavelet transform is another popular linear signal transformation that provides improved time transform space resolution (Improved over the time transform space resolution of the FFT and Yule-Walker AR). The transform space of the continuous wavelet transform is the time scale space representation. Higher scales correspond to lower frequencies and vice versa. The continuous wavelet transform allows one to specify the scales to use during transformation. Furthermore one can choose an arbitrary function to linearly project the time series on. Popular wavelet functions (examples: Mexican hat and Morlet wavelet) are often very similar to the tapered cosine functions used in a windowed short time FFT. The more flexible continuous wavelet transform should provide better classification performance compared to the FFT and Yule-Walker AR methods. However in the experiments done for this study the continuous wavelet transform performed worse or almost similar to the FFT and Yule-Walker AR method. This is probably due to the high number of parameters one can choose/optimize (it is more difficult to find a good set of parameters).

6 Classification

6.1 Introduction

In the previous chapter a possible feature extraction implementation and justification was given for our volcano data set. We did already conclude that the optimized spectrogram representation did not improve classification performance over a single averaged spectrum whilst using the frame order independent kernel density estimation classifier. In this chapter it is of interest to see if there are perhaps better classifiers and classification strategies for the given volcano data-set and feature representation. Thus in this chapter several other classification strategies and techniques are discussed and used on the given volcano data set. In figure 6.1 a categorization of classification techniques is given. Typically the classification block transforms the provided feature representation to a measure of class membership (class label or class posterior probability).

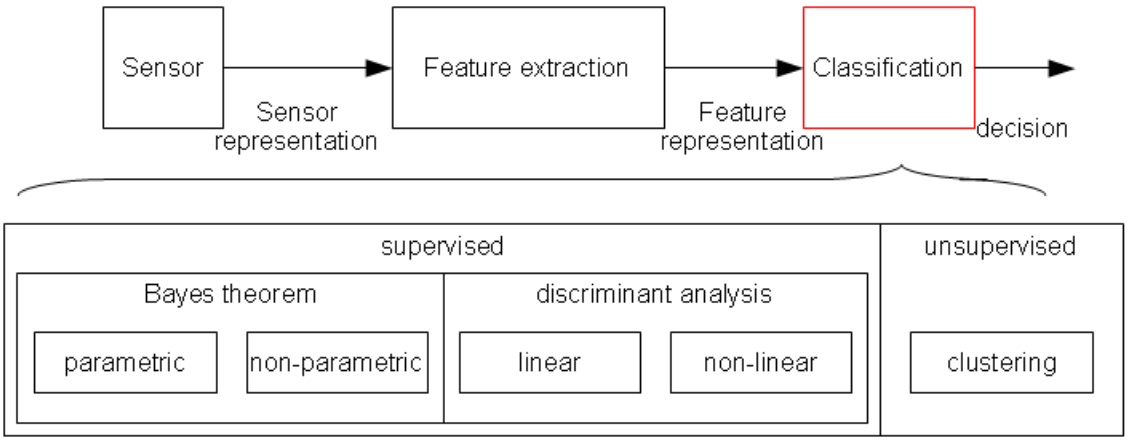

Figure 6.1: Categorization of classification techniques according to [3]

There are two main divisions of classification techniques. The first main classification division is supervised classification. Supervised classification is sometimes also referred to as discrimination. In supervised classification a labelled set of examples is given a priori. This a priori given set of examples is also referred to as the training set. The labels are usually assigned to the training examples by a human expert but the labels could also be assigned to the training examples by an unsupervised classification algorithm. The task of the supervised classification algorithm is to assign class labels to new unseen presented examples using the information from the labelled training set. To achieve this the unsupervised classification algorithm has to generalize from the labelled training examples in a 'reasonable' way.

The second main classification scheme is unsupervised classification. Unsupervised classification is often also referred to as clustering. In unsupervised classification no labelled training examples are present. Unsupervised classification is concerned in finding ‘natural’ groups in the given data based on a priori chosen distance or dissimilarity measure. The choice of the distance or dissimilarity measure is often crucial and determines the resulting group size and shape. Clustering algorithms are also often used as the first initialization for other supervised classification algorithms.

In this study supervised classification techniques are used to assign class labels to new unseen test examples. However some supervised classification algorithms try to find an explicit structure in the provided training set. These supervised classification algorithms often require a reasonable initial grouping of the data. In these cases clustering techniques are used. Clustering techniques are not discussed in this work.

6.2 Supervised classification techniques

The supervised classification scheme is again subdivided into two supervised classification schemes (see again figure 6.1). The first supervised classification scheme is classification using the Bayes theorem. The second supervised classification scheme is discriminant analysis.

6.2.1 Classification using the Bayes theorem

Classification using the Bayes theorem is based upon the knowledge of the class posterior probability density function of each class. The class posterior probability density function $p(\omega_i|\boldsymbol{x})$ quantifies the probability on a particular class ω_i given an observation or measurement $\boldsymbol{x}$.

The Bayes decision rule for minimum error assigns the given observation or measurement $\boldsymbol{x}$ to this class for which the class posterior probability density function is greatest (see equation 6.1). This Bayes decision rule minimizes the probability of making an error (Actually the resulting error from this decision rule is optimal). Other Bayes decision rules exist. For example there is also a Bayes decision rule which minimizes the risk. This decision rule is interesting when the cost associated with misclassification depends upon the true class of the observation and the class to which the observation is assigned. In this study the Bayes decision rule for minimum error is used (each error type is equally weighted).

$$i_{max} = \underset{i=1,2,\ldots,C}{argmax}(p(\omega_i|\boldsymbol{x}))$$

Equation 6.1: The Bayes decision rule for minimum error

In practice the true underlying class posterior probability density functions are seldom known, instead the class posterior density functions are estimated from a training set. The class posterior density functions may be expressed in terms of the prior probabilities $p(\omega_i)$, $p(\boldsymbol{x})$ and the class conditional density functions $p(\boldsymbol{x}|\omega_i)$ (see equation 6.2). This equation is the well known Bayes rule. It is at least very often more natural to express the class posterior density functions in terms of the prior probabilities and the class conditional density functions instead of estimating the class posteriors directly from the available data-set.

$$p(\omega_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_i)\,p(\omega_i)}{p(\boldsymbol{x})}$$

Equation 6.2: The Bayes rule

When using the Bayes decision rule for minimum error it is not required to estimate the observation prior density function from the available data-set because the class ordering is not altered by the class independent observation prior $p(\boldsymbol{x})$. The practical Bayes decision rule for minimum error is given in equation 6.3.

$$p(\omega_i|\boldsymbol{x}) \propto p(\boldsymbol{x}|\omega_i)\,p(\omega_i) \qquad i_{max} = \underset{i=1,2,\ldots,C}{argmax}(p(\boldsymbol{x}|\omega_i)\,p(\omega_i))$$

Equation 6.3: Practical Bayes decision rule for minimum error

Thus when using the Bayes theorem one needs to estimate the class priors and the class conditional density functions. The class priors are often estimated using the empirical class frequencies. Or in other words how often a class occurs in a given training set relative to the size of the training set. One approach to estimate the class conditional density functions is to assume a simple underlying parametric distribution. Often a multivariate normal distribution is assumed. Another approach is to estimate the class conditional density functions using a non-parametric method. When using non-parametric methods no underlying distribution is assumed and the class conditional density functions can be of arbitrary shape. Complexity of the resulting density functions is typically

controlled by one or more smoothing parameters.

6.2.2 Parametric Bayes classifiers

6.2.2.1 Quadratic normal Bayes classifier

The quadratic normal Bayes classifier is a parametric Bayes classifier based upon the normal distribution (The Bayes classifier was already explained in the previous paragraph). When using the quadratic normal Bayes classifier the class conditional density functions $p(\boldsymbol{x}|\omega_i)$ are assumed to be normally distributed (see equation 6.4). In equation 6.4 $\boldsymbol{m}_i$ is the sample mean for class i and Σ_i is the sample covariance matrix for class i. The sample mean and covariance matrix are estimated for each class. For unequal class covariance matrices the resulting decision boundaries between the corresponding classes are quadratic. In equation 6.4 p is the dimensionality of the measurements. The number of parameters one needs to estimate per class grows quadratically with the dimensionality of the measurements.

$$p(\boldsymbol{x}|\omega_i) = \frac{1}{\sqrt{(2\pi)^p det(\Sigma_i)}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^T \Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)\right)$$

$$\boldsymbol{m}_i = \frac{1}{n_i}\sum_{k=1}^{n_i} \boldsymbol{x}_k$$

$$\boldsymbol{\Sigma}_i = \frac{1}{n_i}\sum_{k=1}^{n_i} (\boldsymbol{x}_k-\boldsymbol{m}_i)(\boldsymbol{x}_k-\boldsymbol{m}_i)^T$$

Equation 6.4: Class conditional density function for the quadratic normal Bayes classifier

6.2.2.2 Linear normal Bayes classifier

When using the quadratic normal Bayes classifier problems can occur with the inversion of the per class sample covariance matrices Σ_i. Typically these problems occur when there are to little observations in a high dimensional feature space. An alternative is to reduce the dimensionality of the observations prior to classification. There is a good chance that the inversion of the covariance matrices is not problematic in the reduced feature space. Another alternative is to reduce the complexity of the classifier. The linear normal Bayes classifier assumes that the class covariance matrices Σ_i are all equal. The per class covariance matrices are replaced by one weighted average covariance matrix $\bar{\Sigma}$ (see equation 6.5). The resulting decision boundaries between classes are linear.

$$p(\boldsymbol{x}|\omega_i) = \frac{1}{\sqrt{(2\pi)^p det(\bar{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^T \bar{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)\right)$$

$$\bar{\Sigma} = \sum_{i=1}^{C} \frac{n_i}{n} \Sigma_i$$

Equation 6.5: Class conditional density function for the linear normal Bayes classifier

6.2.2.3 Uncorrelated normal Bayes classifier.

Another possibility to reduce the complexity of the quadratic normal Bayes classifier is to assume uncorrelated features. The resulting per class covariance matrices Σ_i are (effectively) multiplied by the identity matrix $\boldsymbol{I}$ (see equation6.6). Note that this multiplication is an element wise multiplication. The required number of parameters per class now grows linearly with the observation dimensionality. Resulting decision boundaries are quadratic for unequal diagonal covariance matrices.

$$p(\boldsymbol{x}|\omega_i) = \frac{1}{\sqrt{(2\pi)^p det(\Sigma_i * \boldsymbol{I})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^T(\Sigma_i * \boldsymbol{I})^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)\right)$$

Equation 6.6: Class conditional density function for the uncorrelated normal Bayes classifier

6.2.2.4 Scaled nearest mean normal Bayes classifier.

The scaled nearest mean normal Bayes classifier is an even further simplification of the quadratic normal Bayes classifier. This classifier assumes both uncorrelated features and equal class covariance matrices. The resulting decision boundaries are linear. The difference between this classifier and the nearest mean classifier is that this classifier does incorporate class priors and average feature variances. Unequal class priors introduce a translation of the decision boundary. Unequal average feature variances introduce a rotation of the decision boundary. Both in comparison to the nearest mean classifier.

$$p(\boldsymbol{x}|\omega_i) = \frac{1}{\sqrt{(2\pi)^p det(\bar{\Sigma} * \boldsymbol{I})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^T(\bar{\Sigma} * \boldsymbol{I})^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)\right)$$

Equation 6.7: Class conditional density function for the scaled nearest mean normal Bayes classifier

6.2.2.5 Nearest mean normal Bayes classifier.

The nearest mean normal Bayes classifier is the simplest form of the normal Bayes classifier. The assumed class covariance matrices are equal to the scaled identity matrix (see equation 6.8). The scaling is required to avoid numerical precision issues but does not influence the place of the decision boundaries. The scale can be an average dataset variance. Of course the nearest mean classifier can trivially be implemented without using density estimation techniques.

$$i_{max} = \underset{i=1,2,\ldots,C}{argmax}\left(p(\boldsymbol{x}|\omega_i)\right)$$

$$p(\boldsymbol{x}|\omega_i) = \frac{1}{\sqrt{(2\pi)^p det(\sigma^2 \boldsymbol{I})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^T \sigma^2 \boldsymbol{I} (\boldsymbol{x}-\boldsymbol{m}_i)\right)$$

Equation 6.8: Class conditional density function for the nearest mean normal Bayes classifier

The mathematical expressions of the normal Bayes classifiers given in this paragraph can be simplified using log likelihoods and using the Bayes rule for minimum error. For example taking the logarithm over the class conditional density functions does not alter the class ordering. And for all the class conditional density expressions class ordering is independent of the $(2\pi)^p$ normalization part. The less complex forms of the normal Bayes classifier can be simplified even further.

6.2.2.6 Normal mixture Bayes classifier

The simple normal Bayes classifiers discussed in the previous paragraphs perform well when the training and test observations are drawn from a per class uni-modal distribution that is similar to the normal distribution. However the per class distributions are at least not always uni-modal and similar to the normal distribution. In these other circumstances the more flexible Normal mixture Bayes classifier might provide improved classification performance over the simple normal Bayes classifiers. The class conditional density function of the normal mixture Bayes classifier is a weighted summation of normal distributions (see equation 6.9). The weights or mixing proportions π_{ij} should all be positive or zero. Furthermore the sum over all the per class mixing proportions should equal to one. Decision boundaries can be of arbitrary shape depending on the number of normal distributions per class conditional density function. Classifier complexity is mainly controlled by the number of normal distributions. The number of normal distributions used per class is a parameter of interest. Again one can reduce the classifier complexity by assuming uncorrelated features and or equal covariance matrices.

$$p(\boldsymbol{x}|\omega_i)=\sum_{j=1}^{J_i}\pi_{ij}\Re(\boldsymbol{x},\theta_{ij})$$

$$\pi_{ij}\geqslant 0 \quad i=1,\ldots,C \quad j=1,\ldots,J_i \quad \sum_{j=1}^{J_i}\pi_{ij}=1 \quad i=1,\ldots,C$$

$$\Re(\boldsymbol{x},\theta_{ij})=\frac{1}{\sqrt{(2\pi)^p\, det(\boldsymbol{\Sigma_{ij}})}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m_{ij}})^T\boldsymbol{\Sigma_{ij}}(\boldsymbol{x}-\boldsymbol{m_{ij}})\right) \quad \theta_{ij}=[\boldsymbol{m_{ij}},\Sigma_{ij}]$$

Equation 6.9: Normal mixture Bayes classifier

When using the normal mixture Bayes classifier there are three sets of parameters one needs to estimate. The first set of parameters are the numbers of normal distributions per class J_i. Of course one can choose an arbitrary and unequal number of normal distributions per class but in this study an equal number of normal distributions per class was used. The second set of parameters are the mixing proportions π_{ij}. And the third and final set of parameters are the means $\boldsymbol{m_{ij}}$ and covariance matrices Σ_{ij}. The difficulty in finding the parameters is that one does not know to which normal distribution the per class training examples belong to. One usually does not know the underlying structuring (if any) of the normal distributions. The optimization criterion for the parameters of the normal mixture model is the maximum likelihood criterion. One would like to choose the discussed parameters such that the maximum likelihood function (see equation 6.10) is at its global maximum for a given set of per class training examples $\boldsymbol{x_{ik}}$.

$$L(\Theta_i)=\prod_{k=1}^{K_i}\sum_{j=1}^{J_i}\pi_{ij}\Re(\boldsymbol{x_{ik}},\theta_{ij}) \quad \Theta_i=[J_i,\pi_{ij},\theta_{ij}] \quad \theta_{ij}=[\boldsymbol{m_{ij}},\Sigma_{ij}]$$

Equation 6.10: Likelihood function for the normal mixture Bayes classifier

In general it is not possible to solve $\partial L/\partial\Theta_i=0$ explicitly for the parameters of the model. However it is possible to find a local maximum using the expectation maximization algorithm. The expectation maximization algorithm is an iterative algorithm that generates a sequence of parameter estimates. Each parameter estimate has a higher or equal likelihood compared to the previous parameter estimate. If the likelihood for the current parameter estimate equals the likelihood for the previous parameter estimate the sequence has converged to a local maximum. Usually a slightly less computational intensive convergence criterion is used because the EM algorithm often converges very slowly towards the local optimum. During the iterations of the expectation maximization algorithm two steps are repeated. The first step is the expectation step (see equation

6.11). In equation 6.11 γ_{ijk} is the normalized probability of occurrence on the training example $\boldsymbol{x}_{ik}$ for class i and component j given the current set of parameters. The k index is the training example index. The second step is the maximization step (see equation 6.12). During the maximization step the parameters of the normal mixture Bayes classifier are re-estimated based upon the results (γ_{ijk})from the expectation step. The formula's in equation 6.11 and equation 6.12 are straight forward and are pretty easy to understand.

$$\gamma_{ijk} = \frac{\pi_{ij}^{(t)} \Re(\boldsymbol{x}_{ik}, \theta_{ij}^{(t)})}{\sum_{j=1}^{J_i} \pi_{ij}^{(t)} \Re(\boldsymbol{x}_{ik}, \theta_{ij}^{(t)})} = \frac{\pi_{ij}^{(t)} \Re(\boldsymbol{x}_{ik}, \theta_{ij}^{(t)})}{p(\boldsymbol{x}_{ik}|\omega_i^{(t)})} \qquad \theta_{ij}^{(t)} = [\boldsymbol{m}_{ij}^{(t)}, \Sigma_{ij}^{(t)}]$$

Equation 6.11: Expectation step for the normal mixture Bayes classifier

$$\pi_{ij}^{(t+1)} = \frac{1}{K_i} \sum_{k=1}^{K_i} \gamma_{ijk}$$

$$\boldsymbol{m}_{ij}^{(t+1)} = \frac{\sum_{k=1}^{K_i} \gamma_{ijk} \boldsymbol{x}_{ik}}{\sum_{k=1}^{K_i} \gamma_{ijk}}$$

$$\Sigma_{ij}^{(t+1)} = \frac{\sum_{k=1}^{K_i} \gamma_{ijk} (\boldsymbol{x}_{ik} - \boldsymbol{m}_{ij}^{(t+1)})(\boldsymbol{x}_{ik} - \boldsymbol{m}_{ij}^{(t+1)})^T}{\sum_{k=1}^{K_i} \gamma_{ijk}}$$

Equation 6.12: Maximization step for the normal mixture Bayes classifier

The local maximum that is found depends upon the initial initialization of the parameters. Often the k-means clustering algorithm is used to find a reasonable initial set of parameters (The k-means clustering algorithm will not be discussed). A random initialization is also possible but there is a possibility that one or more of the normal components are too far away from the training examples resulting in numerical issues such as singular covariance matrices.

6.2.3 Non parametric Bayes classifiers

Non-parametric Bayes classifiers assume no parametric form of the class conditional density functions $p(\boldsymbol{x}|\omega_i)$. Instead the probability of occurrence on a given observation $\boldsymbol{x}$ is computed using the presence or distance to nearby training examples. The probability $\boldsymbol{P}$ that a single given observation drawn from $p(\boldsymbol{x}|\omega_i)$ falls within a region $\Re$ centered around the observation $\boldsymbol{x}$ is given in equation 6.13.

$$\boldsymbol{P} = \int_{\Re} p(\boldsymbol{x}|\omega_i)\ d\boldsymbol{x}$$

Equation 6.13:

Now let us assume that one does not have one but N_i observations drawn from the same class conditional density function (Again i is the class index). In this case the probability that k of these N_i observations fall in the same region $\Re$ is given by the binomial distribution (see equation6.14).

$$\boldsymbol{P_k} = \binom{N_i}{k} \boldsymbol{P^k} (1-\boldsymbol{P})^{(N_i-k)}$$

Equation 6.14:

It can be shown from the properties of the binomial distribution that the ratio k/N_i has an expected value of $\boldsymbol{P}$. Thus $E[k/N_i] = \boldsymbol{P}$. Furthermore one can also show that when N_i grows to infinity the variance on the ratio k/N_I reduces to zero. Thus $var(k/N_i) = 0 \quad N_i \to \infty$. Thus for a sufficiently large N_i, the ratio k/N_i is a good approximation of (or is similar to) the value $\boldsymbol{P}$.

$$\frac{k}{N_i} \sim \boldsymbol{P}$$

Equation 6.15: $\boldsymbol{P}$ is similar to the ratio k/N_i

Finally let is assume that the region $\Re$ is so small that the class conditional density function $p(\boldsymbol{x}|\omega_i)$ is next to constant in this region. Using this assumption in combination with the earlier found approximation for $\boldsymbol{P}$ and one can find an approximation for the class conditional density function (see eqution 6.16). In equation 6.16 V is the volume enclosed by the region $\Re$.

$$\boldsymbol{P} = \int_{\Re} p(\boldsymbol{x}|\omega_i)\, d\boldsymbol{x} \sim p(\boldsymbol{x}|\omega_i) V \ , \ \frac{k}{N_i} \sim \boldsymbol{P} \to p(\boldsymbol{x}|\omega_i) V \sim \frac{k}{N_i}$$

$$p(\boldsymbol{x}|\omega_i) \sim \frac{k}{N_i V}$$

Equation 6.16: General approximation for the non-parametric class conditional density function

In practice N_i is fixed and corresponds to the number of per class training examples. Accuracy of the class conditional density estimate depends upon the volume of region $\Re$ and the number of required training examples k in this region. Preferably one would like to choose the volume such that it is as small as possible to support the assumption that $p(\boldsymbol{x}|\omega_i)$ is constant. On the other hand one would like the volume to be large enough to enclose at least one training example.

6.2.3.1 k nearest neighbour Bayes classifier

In the previous paragraph an approximation for the non-parametric class conditional density function was given. There are two basic classification approaches one can adopt in using the given conditional density function. The first approach is the k nearest neighbour Bayes classifier. The k nearest neighbour Bayes classifier uses a fixed value for k. The region $\Re$ is spherical and centered around the given observation $\boldsymbol{x}$. The radius of the spherical region is chosen such that the region exactly encloses k nearest neighbours (See also figure 6.2). Thus the resulting volume V is the smallest possible spherical volume that also contains k nearest neighbours and is centered around $\boldsymbol{x}$. The volume V is variable in $\boldsymbol{x}$. Complexity of the class conditional density function is controlled with k. A small value of k results in a complex density function with a lot of detail whereas a large value of k results in a smooth density function with little detail. The k nearest neighbour Bayes classifier can also be implemented directly without the estimation of class conditional density functions. This classifier is referred to as the k nearest neighbour classifier. This discriminant classifier does not incorporate the class priors. In this study euclidean distances are used. Furthermore in this study the value of k is optimized using internal cross-validation on the training set.

$$i_{max} = \underset{i=1,2,\ldots,C}{argmax}\left(p(\boldsymbol{x}|\omega_i)\,p(\omega_i)\right)$$

$$p(\boldsymbol{x}|\omega_i) = \frac{k}{N_i V}$$

$$V_d(R) = C_d R^d \qquad C_d = \frac{\pi^{\frac{d}{2}}}{\Gamma\left(\frac{d}{2}+1\right)}$$

Equation 6.17: k Nearest Neighbour Bayes classifier

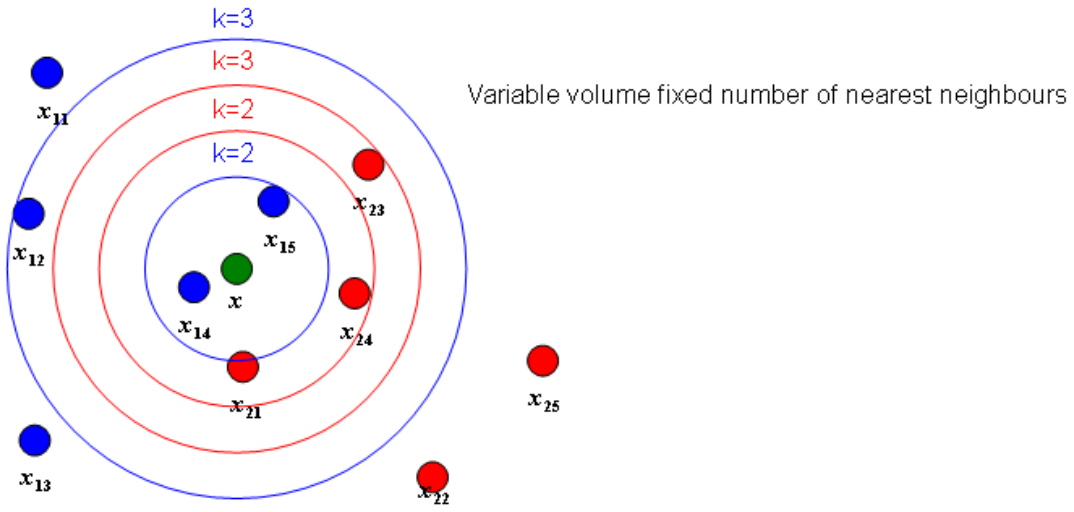

Figure 6.2: The volume V is variable in $\boldsymbol{x}$, the number of nearest neighbours is fixed

6.2.3.2 kernel density Bayes classifiers

The kernel density Bayes classifier was already introduced informally in chapter 5 were we needed a simple and flexible classifier for the feature extraction experiments. The kernel density Bayes classifier is also referred to as the Parzen classifier named after its inventor Emanuel Parzen. The kernel density classifier uses a fixed region volume V and finds the number of training examples within this fixed region. If one would use the non parametric class conditional density function from equation 6.16 directly on would find a density function that is discontinuous in $\boldsymbol{x}$. Every time one of the training examples enters or exits the fixed volume region results in a discontinuity in the density function. This is often not desirable. Therefore equation 6.16 is slightly modified. Instead of counting the training examples within a fixed volume, the distances between the given example $\boldsymbol{x}$ and all the training examples $\boldsymbol{x}_{ik}$ are computed. These distances in turn are weighted using a kernel. More distant training examples typically receive a smaller value from the kernel compared to nearby training examples. Note that the volume of the fixed region is now controlled by the choice of the kernel function. The algorithm incorporates all the training examples but the kernel decides if a training example is inside a given finite volume. Again the kernel density classifier is given in equation 6.18. In this study the multivariate normal kernel was used.

$$i_{max} = \underset{i=1,2,\ldots,C}{argmax}\left(p(\boldsymbol{x}|\omega_i)\,p(\omega_i)\right)$$

$$p(\boldsymbol{x}|\omega_i) = \frac{1}{nh^p}\sum_{k=1}^{K_i} K\left(\frac{1}{h}(\boldsymbol{x}-\boldsymbol{x}_{ik})\right)$$

$$K(z) = \frac{1}{(2\pi)^{p/2}}\exp\left\{-\frac{z^T z}{2}\right\}$$

Equation 6.18: Kernel density Bayes classifier

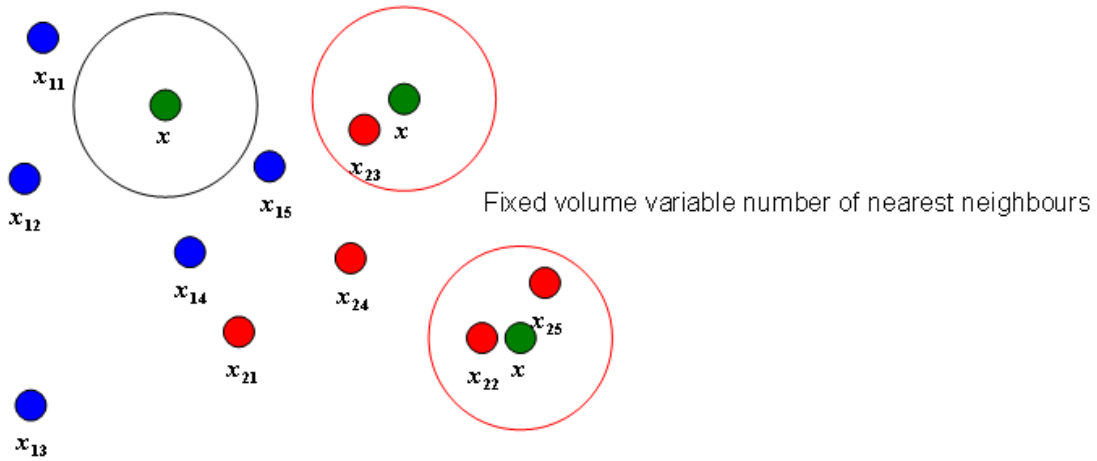

Figure 6.3: The volume V is fixed the number of nearest neighbours is variable

6.2.4 Hidden Markov models

Introduction
The Bayes classifiers discussed up until so far are able to classify objects based upon a single feature or measurement vector. Or when a sequence of feature vectors is given for each object one can compute the class conditional probability for each feature vector independently and combine the resulting probabilities using the product rule to find the probability on the sequence. The typical seismic recording is the result of registering several different seismic wave types and is therefore not stationary. The order of arrival of the different wave types is approximately fixed. The time between the arrival of different wave types can vary depending on the distance between the hypocentrum and the point of measurement. Therefore the order of the feature vectors inside the recordings might be of importance.

Markov models and the Markov property
Hidden Markov models are a widely used method for sequence modelling. A hidden Markov model is a statistical model for which the underlying system that is modelled is assumed to be a Markov random process. A Markov random process is a time varying random process for which the Markov property holds. A given random process has the Markov property if the conditional probability distribution of the future state only depends upon the present state and not on the past states. Thus the probability on the future process state is conditional independent of the past process states given the current process state (see equation 6.19). In equation 6.19 $Q_n, \ldots, Q_0$ are the random process state variables taking on the state values $q_n, \ldots, q_0$ for the sequence indices $n, \ldots, 0$.

$$p(Q_n = q_n | Q_{n-1} = q_{n-1}, \ldots, Q_0 = q_0) = p(Q_n = q_n | Q_{n-1} = q_{n-1})$$

Equation 6.19: The Markov property

Continuous and discrete Markov models
Both continuous and discrete state space Markov models exist (continuous state space Markov models through the use of Harris chains). In this study we assumed that the seismic recordings are the result of a countable number of underlying physical processes. There are at least a countable number of wave types and perhaps also a countable number of states inside each earthquake type. Therefore discrete state space Markov models were used in this study. Spectral characteristics are measured at a finite number of positions inside each time discrete seismic recording. Therefore in this case it is most natural to use a time discrete Markov model. A Markov model that has both a discrete state space as well as discrete sequence indexing is also referred to as a Markov chain.

Hidden Markov models
For a Markov model the state and state sequence of the model are directly observable through the outputs of the model. The only parameters of interest are the state transition probabilities a_{ij} and possibly the state prior probabilities π_i. The state transition probabilities are almost always organized in a square state transition matrix. The sum over the destination state transition probabilities for each source state should equal to one. For a hidden Markov model the state and state sequence are not directly observable from the model outputs. For a hidden Markov model each state has its own probability distribution over the possible outputs of the model. These probability distributions are also referred to as the state emission probability distributions $b_i(\boldsymbol{x})$. Thus a typical hidden Markov model consist of three set of parameters: the state prior, the state transition and the state emission probabilities (see also figure 6.4). In this study both the state prior and state transition probability distributions are discrete. Furthermore in this study both continuous and discrete state emission probability distributions are used. Continuous state emission probabilities in the form of a normal mixture model for each state. And discrete emission probabilities in the form of a histogram of output probabilities for each state. From now on when referring to continuous

hidden Markov models, continuous emission density hidden Markov models are meant. When referring to discrete hidden Markov models, discrete emission density hidden Markov models are meant.

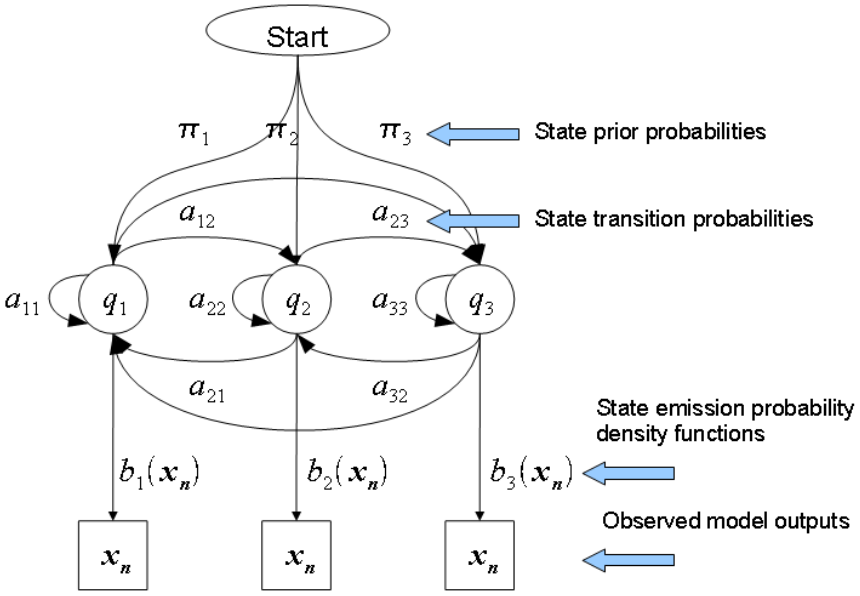

Figure 6.4: Typical (fully connected) hidden Markov with three states

Hidden Markov model problems
There are three basic problems associated with hidden Markov models:

- Given the model parameters A, B and Π, it is of interest to be able to compute the probability on a given observed output sequence $x_1, \ldots, x_n$. The probability on a given observed output sequence is the sum over all possible state sequences. The sum over all possible state sequences is a combinatorial explosion. For the direct/naive implementation the number of required computations grows exponentially with the sequence length and is impractical for all but the simplest hidden Markov models in combination with very short observation sequences. Luckily an efficient dynamic programming algorithm exists namely the forward algorithm[18][19]. For the forward algorithm the number of required computations is linear in the sequence length and quadratic in the number of states.

- Given the model parameters and a given observed output sequence $x_1, \ldots, x_n$ it is of interest to find a corresponding state sequence $q_1, \ldots, q_n$ which is the most likely to have generated the observed output sequence. The direct/naive implementation enumerates over all possible state sequences and picks this sequence for which the probability of generating the output sequence is highest. Again a dynamic programming algorithm exists which solves this problem efficiently. This algorithm is also referred to as the Viterbi algorithm[18][19].

- The last problem is to find a set of parameters for a hidden Markov model which maximizes the likelihood on one or multiple output sequences. This problem is typically solved using the well known Baum-Welch algorithm[18][19]. Which is a special version of the expectation maximization algorithm.

Hidden Markov model problems in this study
In this study both problem 1 and problem 3 are of interest. First one would like to find/train a hidden Markov model for each seismic signal class based upon a set of training sequences. Second one would like to classify new unseen test sequences in their appropriate seismic signal class. In this study the Baum-Welch training algorithm is used to train a hidden Markov model for each

signal class. The number of states used for each hidden Markov model is a parameter of interest. The forward algorithm is used to compute the class conditional probabilities on the output sequences. The Bayes decision rule for minimum error is used to decide on the signal class. Hidden Markov models are more complicated compared to the Bayes classifiers discussed up until so far. Describing the mathematical details of the hidden Markov models would go beyond the scope of this report therefore I will not discuss the mathematical details of the hidden Markov models. Instead I would like to refer to [18] and [19] which are both excellent tutorials on hidden Markov models.

6.2.5 Discriminant analysis

The general idea of supervised discriminant analysis is to find a linear or non-linear combination of features (the discriminant function) which best characterizes or separates two or more classes of observations (see figure 6.5). In this paragraph we will look briefly at the two class versions of a couple of discriminant analysis classifiers. Note that the generalization of the described classifiers towards multi class problems is often not trivial (This in contrast to the Bayes classifiers). But an elaborate discussion of the multi class discriminant analysis classifiers would go beyond the scope of this report.

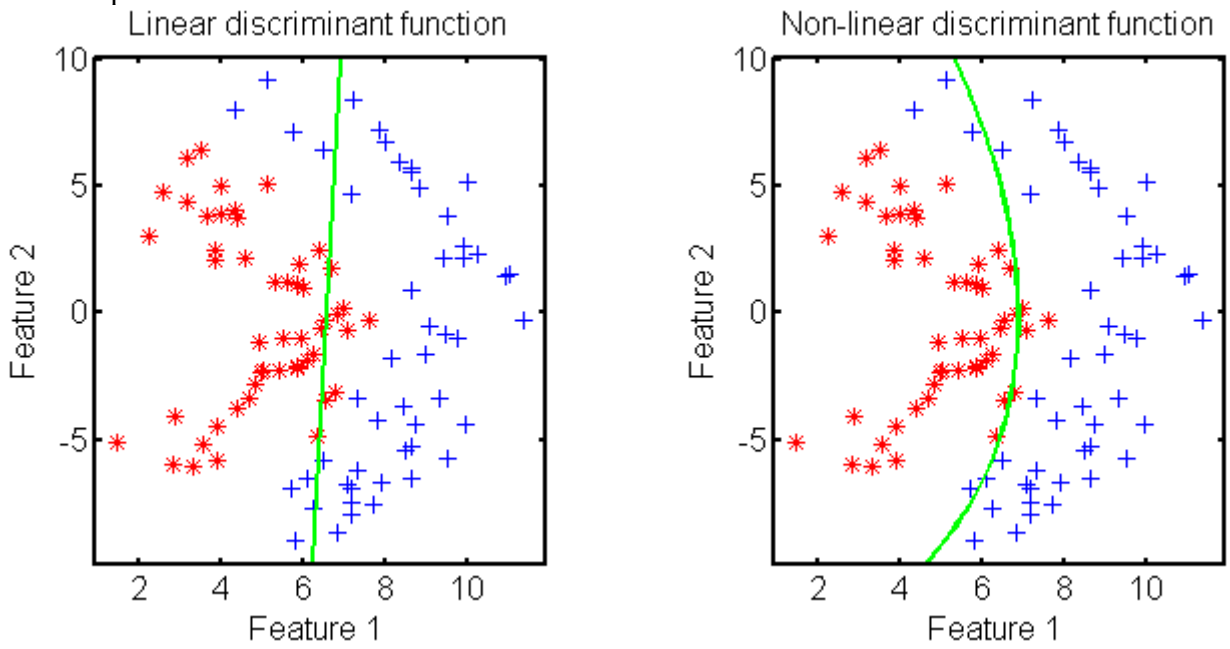

Figure 6.5: Linear discriminant function (left) Non-linear discriminant function (right)

6.2.5 Linear discriminant analysis

Two class linear discriminant classifiers use a linear discriminant function of the form given in equation 6.20. The goal of these classifiers is to find a suitable weighted combination of the features $\boldsymbol{w}$ and a threshold or offset weight w_0 . When the weighted combination of a given feature vector $\boldsymbol{x}$ is larger than zero then it is assigned to the first class. If the weighted combination is smaller than zero the given feature vector is assigned to the second class (see again equation 6.20).

$$\boldsymbol{w}^T\boldsymbol{x}+w_0 \begin{cases} >0 \\ <0 \end{cases} \Rightarrow \boldsymbol{x} \in \begin{cases} \omega_1 \\ \omega_2 \end{cases}$$

Equation 6.20: Two class linear discriminant classifier

6.2.5.1 Nearest mean classifier

The nearest mean classifier separates the two classes of objects by a linear discriminant function that is perpendicular to the line through the class means $\boldsymbol{m_1}$ and $\boldsymbol{m_2}$. The discriminant function is placed exactly in between the two class means. The class priors are ignored by this classifier. (see equation 6.21) The implementation used for this study scales the nearest mean discriminant function such that the posterior probabilities of the training examples are maximized (not shown in equation 6.21).

$$\boldsymbol{w}=\boldsymbol{m}_1-\boldsymbol{m}_2$$
$$w_0=\frac{1}{2}\left(\boldsymbol{m}_2{}^T\boldsymbol{m}_2-\boldsymbol{m}_1{}^T\boldsymbol{m}_1\right)$$

Equation 6.21: Discriminant function for the two class nearest mean classifier

6.2.5.2 Scaled nearest mean classifier

The scaled nearest mean classifier is already more complicated compared to the nearest mean classifier. The classes are separated by a discriminant function that is possibly a rotated and translated version of the nearest mean discriminant function. The possible relative rotation of the scaled nearest mean discriminant function depends on $\hat{\Sigma}$ (see equation 6.22). $\hat{\Sigma}$ is a class prior weighted sum of class covariance matrices followed by an element wise multiplication with the identity matrix. The possible translation of the discriminant function depends on the log ratio of the class priors. The implementation used for this study scales the scaled nearest mean discriminant function such that the posterior probabilities of the training examples are maximized

$$\hat{\Sigma} = (p(\omega_1)\hat{\Sigma}_1 + p(\omega_2)\hat{\Sigma}_2) * \boldsymbol{I}$$

$$\boldsymbol{w} = \hat{\Sigma}^{-1}(\boldsymbol{m_1} - \boldsymbol{m_2})$$

$$w_0 = \frac{1}{2}(\boldsymbol{m_2}^T \hat{\Sigma}^{-1} \boldsymbol{m_2} - \boldsymbol{m_1}^T \hat{\Sigma}^{-1} \boldsymbol{m_1}) + \log\left(\frac{p(\omega_1)}{p(\omega_2)}\right)$$

Equation 6.22: Discriminant function for the two class scaled nearest mean classier

6.2.5.3 Fisher classifier

The two class fisher classifier is given in equation 6.23. The two class fisher classifier is mathematically very similar to the two class scaled nearest mean classifier. The only difference is that the fisher classifier uses the full within class covariance matrix whereas the scaled nearest mean classifier only uses the diagonal within class covariance matrix. The two class fisher classifier is the result of maximizing the fisher criterion (not given here). The fisher criterion is defined as the ratio of the between class and within class variances.

$$\hat{\Sigma} = p(\omega_1)\hat{\Sigma}_1 + p(\omega_2)\hat{\Sigma}_2$$

$$\boldsymbol{w} = \hat{\Sigma}^{-1}(\boldsymbol{m_1} - \boldsymbol{m_2})$$

$$w_0 = \frac{1}{2}(\boldsymbol{m_2}^T \hat{\Sigma}^{-1} \boldsymbol{m_2} - \boldsymbol{m_1}^T \hat{\Sigma}^{-1} \boldsymbol{m_1}) + \log\left(\frac{p(\omega_1)}{p(\omega_2)}\right)$$

Equation 6.23: Discriminant function for the two class fisher classifier

6.2.5.4 Linear perceptron classifier

The linear perceptron classifier tries to minimize the perceptron criterion function given in equation 6.24. The perceptron criterion function is proportional to the sum of distances of the misclassified samples to the decision boundary. The perceptron criterion function is thus only based upon the misclassified samples. One would like to find a discriminant function for which the perceptron criterion is minimized. Because the perceptron criterion function is continuous in the discriminant function one can find a solution using an iterative gradient based procedure (not given here). When the training examples are linearly separable, then the gradient based procedure is guaranteed to converge to a solution that separates the classes. When the classes are not linearly separable, then the solution will oscillate and no convergence occurs. The often used solution is to use a decreasing training step size whilst using the gradient based procedure. For very large training iteration indices the training step size approaches zero ensuring convergence in all cases.

$$J_P(\boldsymbol{w}, w_0) = \sum_{\boldsymbol{x}_i \in \boldsymbol{X}_2} \boldsymbol{w}^T \boldsymbol{x}_i + w_0 \; - \sum_{\boldsymbol{x}_i \in \boldsymbol{X}_1} \boldsymbol{w}^T \boldsymbol{x}_i + w_0$$

$$\boldsymbol{X}_1 = Misclassified \;\; examples \in \omega_1$$

$$\boldsymbol{X}_2 = Misclassified \;\; examples \in \omega_2$$

$$[\hat{\boldsymbol{w}}, \hat{w}_0] = \underset{\boldsymbol{w}, w_0}{argmin}(J_P(\boldsymbol{w}, w_0))$$

Equation 6.24: The linear perceptron classifier is the result op minimizing the perceptron criterion.

6.2.5.5 Logistic classifier

The two class logistic classifier assumes that the linear discriminant function is equal to the logarithm over the ratio of the underlying class posterior probability density functions (see equation 6.25). This assumption allows one to express the class posterior probability density functions in terms of the linear discriminant function. See also equation 6.25. The class posterior probability density functions are logistic functions. Of-course the discussed assumption is not justified for all class distributions. But many real data sets are close to normally distributed and including classes often have similar shapes. In these circumstances the assumption is (a proximately) justified. The parameters $\boldsymbol{w}$ and w_0 of the linear discriminant function are usually found using an iterative optimisation scheme of the maximum likelihood function. The linear discriminant function parameters are chosen such that the product of the class conditional probabilities of the training set are maximized.

$$\log\left(\frac{p(\omega_1|\boldsymbol{x})}{p(\omega_2|\boldsymbol{x})}\right) = \boldsymbol{w}^T \boldsymbol{x} + w_0 \qquad p(\omega_1|\boldsymbol{x}) = \frac{e^{\boldsymbol{w}^T \boldsymbol{x} + w_0}}{1 + e^{\boldsymbol{w}^T \boldsymbol{x} + w_0}} \quad p(\omega_2|\boldsymbol{x}) = \frac{1}{1 + e^{\boldsymbol{w}^T \boldsymbol{x} + w_0}}$$

Equation 6.25: Logistic classifier assumption (left) Resulting class posterior density functions (right)

6.2.5.6 Support vector classifier

The last linear classifier discussed and used in this study is the support vector classifier. The complexity of the decision boundaries of the support vector classifier are or can be controlled by the kernel choice. In this study a linear kernel was used which is equivalent to not using a kernel. The resulting decision boundaries are linear. The idea of the support vector classifier is that one would like to find a linear discriminant function which separates the classes with the largest possible margin. The margin as a function of the linear discriminant function is dependent on the nearest training examples of each class (see equation6.26). The nearest training examples are also referred to as the support vectors. The requirement of separability and placement of the decision boundary in the middle of the margin are both incorporated by a constraint (see again equation 6.26). The constraint requires that the training examples of class ω_1 are all at least a relative distance of one separated from the decision boundary. The training examples of class ω_2 are all required to be at least a relative distance of minus one separated from the decision boundary. The choice for the constant of one is arbitrary. The described constraint optimisation problem is typically solved using the generalization of the Lagrange multipliers (Karush-Kuhn-Tucker conditions) which can also take inequality constraints into account (not given here).

$$\rho = \underset{\boldsymbol{x}_i \in \omega_1}{min}\left(\frac{\boldsymbol{w}^T \boldsymbol{x}_i + w_0}{\|\boldsymbol{w}\|}\right) - \underset{\boldsymbol{x}_i \in \omega_2}{max}\left(\frac{\boldsymbol{w}^T \boldsymbol{x}_i + w_0}{\|\boldsymbol{w}\|}\right) \qquad \begin{matrix} \boldsymbol{w}^T \boldsymbol{x}_i + w_0 \geqslant 1 & \boldsymbol{x}_i \in \omega_1 \\ \boldsymbol{w}^T \boldsymbol{x}_i + w_0 \leqslant -1 & \boldsymbol{x}_i \in \omega_2 \end{matrix}$$

$$[\hat{\boldsymbol{w}}, \hat{w}_0] = \underset{\boldsymbol{w}, w_0}{argmax}(\rho(\boldsymbol{w}, w_0))$$

Equation 6.26: Optimization function maximum margin (left) Constraint function (right).

6.2.5.7 Quadratic classifier

The two class quadratic discriminant classifier uses a quadratic discriminant function of the form given in equation 6.27. The quadratic classifier is equal to the fisher classifier when the within class covariance matrices $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$ are equal. Similar to the quadratic Bayes classifier the inversion of the class covariance matrices can become problematic when an insufficient number of feature vectors is available. Although the formula given in equation 6.27 is a good and often used solution, other solutions for finding a quadratic classifiers exist. One such method is to create a new and longer measurement vector from the old feature vector by augmenting all pairwise products of individual features or measurements. Finding a quadratic classifier for the original feature vectors is now equal to finding a linear classifier for the pairwise product expanded feature vectors. This method is also referred to as the kernel trick.

$$\boldsymbol{x}^{\boldsymbol{T}} \boldsymbol{W}\, \boldsymbol{x} + \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{x} + w_0$$

$$\boldsymbol{W} = \hat{\Sigma}_2^{-1} - \hat{\Sigma}_1^{-1}$$
$$\boldsymbol{w} = 2(\boldsymbol{m_1}^T \hat{\Sigma}_1^{-1} - \boldsymbol{m_2}^T \hat{\Sigma}_2^{-1})$$
$$w_0 = \boldsymbol{m_2}^T \hat{\Sigma}_2^{-1} \boldsymbol{m_2} - \boldsymbol{m_1}^T \hat{\Sigma}_1^{-1} \boldsymbol{m_1} + \log\left(\frac{det(\hat{\Sigma}_2)}{det(\hat{\Sigma}_1)}\right)$$

Equation 6.27: Discriminant function for the two class quadratic classifier

6.3 Experimental setup

6.3.1 Introduction

In chapter 5 the class of transformations for the feature extraction block used in this study was discussed. A justification was given for a reasonable set of block and block parameters. In the coming experiments the FFT will be used in combination with the DCT. Furthermore the Hamming window will be used and the resulting spectra will be scaled using the log transformation. If not otherwise mentioned a window length of 256 will be used in combination with a window overlap of 75%. And finally the first four DCT features are used. The two questions that come to mind for the coming experiments are:

- What are good classifiers for our resulting feature representation?
- What are good classification strategies?

6.3.2 Testing criteria

In these experiments the same testing criteria was used compared to the experiments of chapter 5 but are briefly repeated for the comfort of the reader. In these experiments the holdout estimate was used. The holdout estimate was repeated ten times each with a different random permutation of the volcano data. The ten resulting classification performances were averaged reducing the performance estimate variance and bias. The Mersenne twister random number generator was used in creating the random permutations of the data. And prior to each experiment the random number generator was seeded with the same seed that was also used for all the experiments in chapter 5. The k-means clustering algorithm was used to find a reasonable initial set of normal distributions for the normal mixture Bayes classifier and the hidden Markov models. The k-means clustering algorithm also uses the random Number generator. Therefore the seed of the random number generator was loaded prior to- and stored after each permutation operation ensuring equal dataset permutations for all the experiments.

6.3.3 Data set

In these experiments the same data set was used that was also used for the experiments of chapter 5. The first 133 seismic recordings were selected from each class. Thus a total of 532 seismic recording were used for all the experiments. All classifiers were trained using 100 randomly chosen recordings. Classification performance was tested on the remaining 33 recordings. Finally for these experiments only the recordings originating from the OLL station were used. In the next chapter we will look at combined classification results using the recordings of all the stations.

6.3.4 Classifiers

Most of the classifiers used for these experiments were already discussed in the previous paragraphs. Thus a detailed explanation is not given again. However an enumeration of the classifiers used in these experiments is given in table 6.1 When applicable classifiers are optimized on each presented training set. For example internal cross validation on the training set was used to find the optimal number of nearest neighbours when using the nearest neighbour classifier.

Name:	Abbreviation:	Decision boundary:	Optimization parameter:
Parametric normal Bayes classifiers			
Linear normal Bayes classifier	ldc.	Linear	-
Uncorrelated normal Bayes classifier	udc.	Quadratic	-
Quadratic normal Bayes classifier	qdc.	Quadratic	-
Normal mixture Bayes classifier	mogc.	Arbitrary(1)	mixtures
Continuous hidden Markov classifier	-	Arbitrary(2)	states/mixtures
Non parametric normal Bayes classifiers			
Kernel density Bayes classifier (h)	parzenc.	Arbitrary(3)	h
Kernel density Bayes classifier (h1,...,hc)	parzendc.	Arbitrary(3)	h1,...,hc
Discrete hidden Markov classifier	-	Arbitrary(4)	states/symbols
Linear discriminant classifiers			
Nearest mean Bayes classifier	nmc.	Linear	-
Scaled nearest mean Bayes classifier	nmsc.	Linear	-
Linear perceptron classifier	perlc.	Linear	-
Fisher classifier	fisherc.	Linear	-
Logistic classifier	loglc.	Linear	-
Support vector classifier	svc.	Linear(5)	-
Non linear discriminant classifiers			
Quadratic classifier	qdc.	Quadratic	-
k-nearest neighbour classifier	knnc.	Arbitrary(6)	k

1 Depending on the number of normal distributions
2 Depending on the number of normal distributions and states
3 Depending on the smoothing parameter(s)
4 Depending on the number of histogram bins
5 Depending on the distance measure
6 Depending on the number of nearest neighbours

Table 6.1: Classifiers used in the discussed experiments

6.3.5 Classification strategies

Use only frequency information:
For the first classification strategy a spectrogram was computed for each segmented seismic recording (See also figure 6.6). The resulting spectrogram was averaged in the time/frame direction resulting in one averaged spectrum. This averaged spectrum in turn was scaled using the log (dB.) transformation and finally the dimensionality was reduced using the DCT. The dimensionality of the resulting feature vectors is four and each recording is represented by only one feature vector. Only (dimensionality reduced and decorrelated) frequency information is present in this representation. The complexity of the spectra is controlled by the window length which is a parameter of interest. The window overlap was set to one minus the window length.

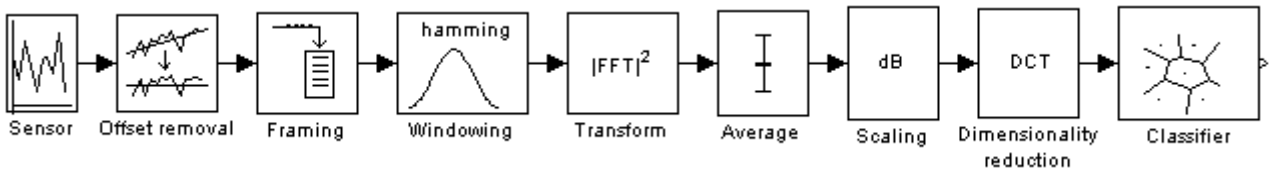

Figure 6.6:

Use time frequency information and independent frames:
This strategy was also used in chapter 5 to find a good set of feature extraction blocks and block parameters using the kernel density classifier. Similar to the previous strategy a spectrogram was computed for each recording. The resulting spectral frames for each recording were scaled using the log (dB.) transformation and the dimensionality was reduced using the DCT (see figure 6.7). All frames within a given recording were assumed to come from the same probability distribution. One class conditional density function was estimated on the training set for each class. The frame probabilities were computed independent of each other using one conditional density function per class. The probability on a single test recording is the product of the assumed independent frame probabilities. To avoid numerical issues log probabilities were used. Only the Bayes classifiers were used for this experiment. Both time and frequency information is present in this representation but the ordering of the frames resulting from the seismic recordings is of no importance.

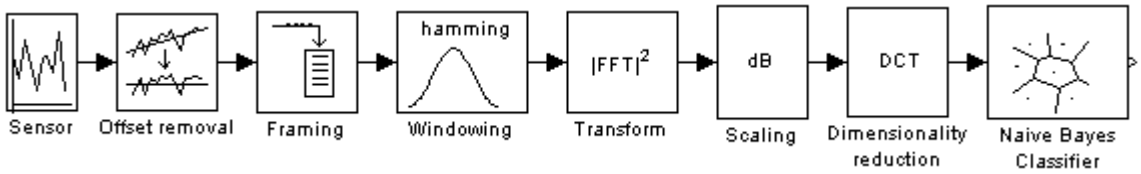

Figure 6.7:

Use time frequency information and hidden Markov models:
For the experiments that use this classification strategy a hidden Markov model was constructed for each class. In contrast to the naive Bayes classifiers used in the previous strategy, hidden Markov models can model the ordering of the frames within the given recordings if present. The class conditional probabilities were computed using the trained hidden Markov models and the well known forward algorithm . A single class conditional probability was computed for each recording. Again the Bayes decision rule for minimum error was used to decide on the class labels. Two types of hidden Markov models were used. The first type of hidden Markov model is the continuous hidden Markov model. The continuous hidden Markov models in these experiments only use one normal mixture function per state. It is possible to use several mixture components per state, but in the experiments done for this study it proofed very difficult to train hidden Markov models with more than one mixture component per state even when uncorrelated covariance matrices were used. Thus for the experiments using the continuous hidden Markov models, classifier complexity was controlled by the number of states only. Full covariance matrices were used for each state. The second type of hidden Markov model used in this study is the discrete hidden Markov model. The discrete hidden Markov models uses a n by m emission histogram. n is the number of states and m is the number of possible observation symbols. For the discrete hidden Markov model complexity was controlled by the number of states and the number of symbols. Both the continuous and discrete hidden Markov models use the k-means clustering algorithm to find a reasonable initial clustering of the data. For the continuous hidden Markov model this initial clustering is not fixed. The normal mixture components can still change shape and move around through the feature space during training. The symbols assigned to the features whilst using the discrete hidden Markov models are fixed and do not change during training. The well known and popular Baum Welch training algorithm was used to train the hidden Markov models. Like the expectation maximization algorithm the Baum welch training algorithm only finds a local maximum therefore the training

procedure was repeated ten times and the best model was used to test classification performance. The average likelihood on the training set was used to select the best hidden Markov model. (Higher likelihoods are better) In these experiments the Baum Welch training runs were stopped after 1000 iterations or when the convergence criterion on the likelihood function was met (likelihood does not increase more than 1.0e-10). Whichever came first. A constant and uniform initialization was used for the state prior and state transition matrices. Scaling of the forward and backward probabilities was used to avoid numerical under-flows[18] for long sequences of small probabilities. One could also use log probabilities [22] to avoid numerical under-flows but the log and exp functions required for the logsum operator are to computationally expensive.

Training hidden Markov models is a time consuming business. A highly optimized c implementation was developed for these experiments both for the continuous as well as the discrete emission density hidden Markov models.

The hidden Markov models used in the experiments could not model the frame ordering whilst using the feature representation used up until so far (See the results section). Complete recordings are too far apart from each other in the feature space. Therefore a second feature representation was introduced based upon the feature differences. In this feature representation the placement of the complete recordings in the feature space is of no importance. The recordings are all placed on top of each other in the new feature space. Only the displacements of neighboring frames in the recordings is of importance.

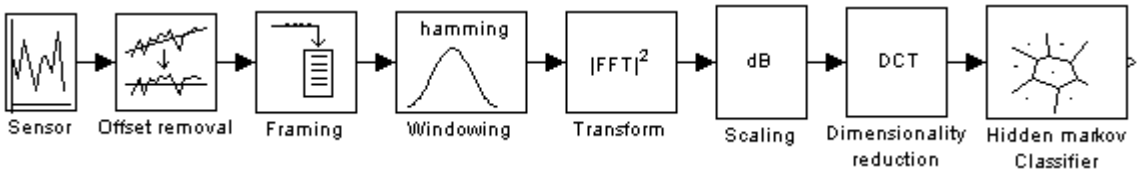

Figure 6.8: hidden Markov model and the original feature space

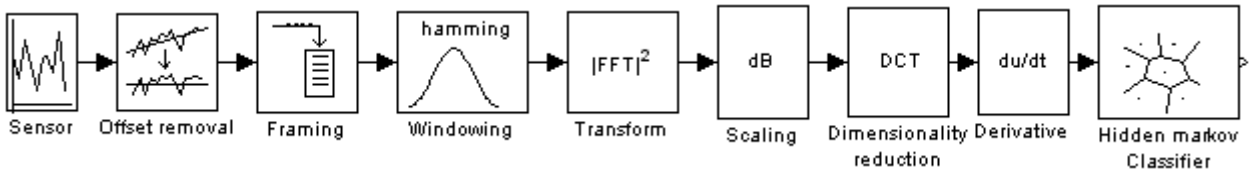

Figure 6.9: hidden Markov model and the derivative or delta feature space

Use time frequency information and concatenated frames:
For this strategy again a very detailed spectrogram was computed for each recording using a window overlap of one minus the window length (see figure 6.10). The detailed spectrogram was divided into a given number of approximately equal sized blocks. The number of blocks is a parameter of interest and fixed for all seismic recordings. The spectral frames in each block were averaged in the time/frame direction. The resulting averaged frames were scaled using the dB. conversion and the dimensionality was reduced using the DCT. The dimensionality reduced spectral frames were concatenated into one long feature vector. The length of the feature vector equals the product of the number of DCT coefficients used and the number of blocks. Thus each seismic recording is summarized in one long feature vector. This single feature vector contains both time and frequency information. This in contrast to the first strategy were the resulting feature vectors only contain frequency information. Parts of the seismic recordings that correspond to the same underlying physical processes such as the arrival of a P-wave should ideally end up at the same location in the resulting feature vectors. There is some offset or displacement tolerance because one is typically averaging over several spectra inside each block. Displacement tolerance decreases with an increasing number of blocks.

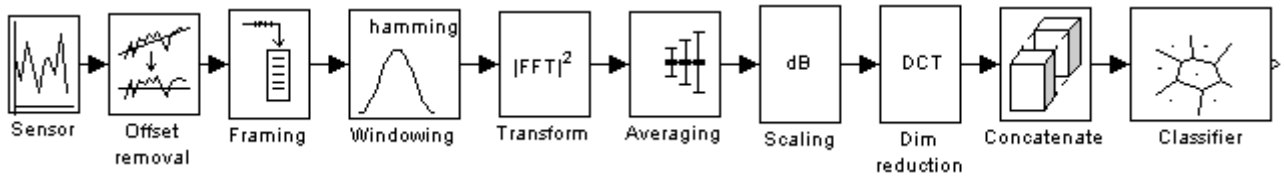

Figure 6.10:

Use time frequency information concatenated frames in the dissimilarity space:
For the final classification strategy the same approach was followed compared to the previous strategy. However instead of using the concatenated feature vectors directly for classification, this strategy first computes the distances or dissimilarities between the feature vectors. These distances or dissimilarities in turn were used for classification. Thus whilst using this strategy classification was done in the dissimilarity space instead of in the feature space. The dimensionality of the dissimilarity space was controlled by the number of examples used in the representation set. The number of examples in the representation set is a parameter of interest. A higher number of examples in the representation set results in a higher dissimilarity dimensionality. When using a distance or dissimilarity representation one also needs to decide on a distance measure. In this study two distance measures were used. The first distance measure used in this study is the well known euclidean distance (see equation 6.28). Whilst using the euclidean distance measure the ordering of the frames inside the concatenated feature vectors is (extremely) important. If one would change the ordering of the frames inside a given feature vector the resulting feature vector most likely ends up in a completely different part of the (high dimensional) feature space and the resulting euclidean distances to other unaltered feature vectors would also be different. Thus just like in the previous strategy the alignment of the time series is of importance whilst using the euclidean distance measure. The second distance measure used in this study is the modified Hausdorff distance (see equation 6.29). The ordering of the frames inside the feature vectors is of no importance for the modified Hausdorff distance. When underlying physical processes produce similar averaged frames in different recordings, then these averaged frames do not have to be exactly aligned in the different resulting feature vectors to achieve a small distance measure.

The representation set was drawn from the training set in a linear and predictable manner. For example when using 200 examples (50 per class) in the representation set all odd numbered examples are drawn from the training set and used in the representation set. The representation examples were always chosen such that an equal number of examples for all classes was included in the representation set.

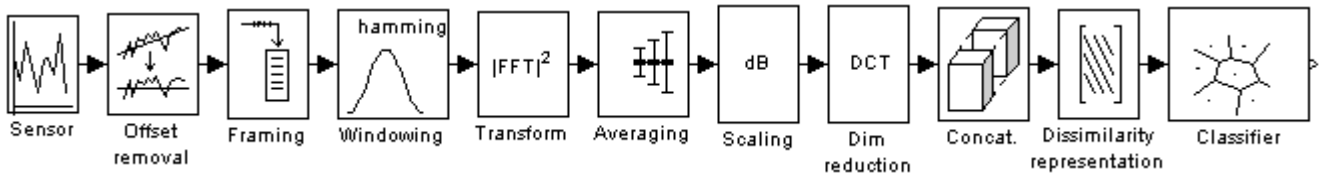

Figure 6.11:

$$d_e(\boldsymbol{x},\boldsymbol{y})=\sqrt{\sum_{i=1}^{p}(x_i-y_i)^2}$$

Equation 6.28: Euclidean distance between (feature) vectors x and y

$$d_h(\boldsymbol{x},\boldsymbol{y})=\frac{1}{M}\sum_{\boldsymbol{a}\in\boldsymbol{x}}\min_{\boldsymbol{b}\in\boldsymbol{y}}\|\boldsymbol{a}-\boldsymbol{b}\|$$

Equation 6.29: Modified Hausdorff distance between (feature) vectors x and y (a and b correspond to the frames inside the concatenated feature vectors and M correspond to the number of frames)

6.4 Results

6.4.1 Classification performance using only frequency information

In figure 6.12 Classification performance is given for several types of classifiers using a single averages spectrum per segmented seismic recording. On the horizontal axis the window length is given. And on the vertical axis the classifier type is given. The classifier names are abbreviated because of limited space (see table 6.1). In this experiment the kernel density classifier with one smoothing parameter per class (parzendc) performed best. Followed by the kernel density classifier with one smoothing parameter for all the classes (parzenc). Again smoothing parameters were optimized using likelihood cross validation. The linear normal Bayes classifier (ldc) and the quadratic normal Bayes classifier (qdc) performed equally well whereas the uncorrelated normal Bayes classifier (udc) performed consistently worse. This indicates that the features of the classes are correlated. The classification results of the nearest mean classifier (nmc) and the scaled nearest mean classifier (nmsc) indicate that the orientation of the decision boundaries relative to the line through the class means is of importance. The normal mixture Bayes classifier (mogc) performed well using shorter window lengths indicating that the class distributions resulting from the shorter window lengths are at least not exactly uni modal normal distributed. But the expectation maximization algorithm used for this experiment failed to converge for the longer window lengths. This is strange because the number and the dimensionality of the training examples remained fixed. Perhaps the resulting class distributions become closer to a uni modal normal distribution causing one of the two mixture covariance matrices to becomes singular. Overall classification performance improves when using more complex classifiers. This is true both for the Bayes classifiers as well as the discriminant analysis classifiers. Again classification performance was hardly dependent on the window length. The experiment with the variable number of DCT features was also repeated. Again classification performance did not improve for higher DCT feature dimensionalities. The optimization algorithm of the linear perception classifier (perlc) failed to find a good discriminant function.

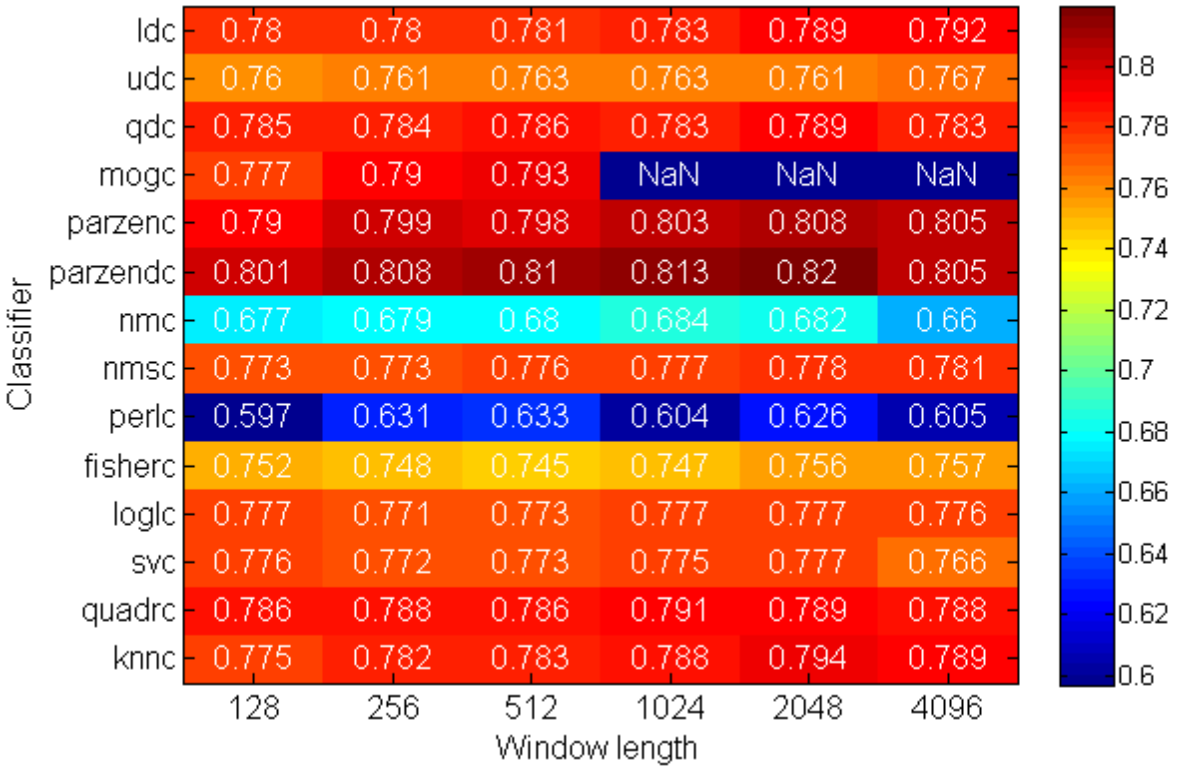

Figure 6.12: Classification performance using only frequency information

6.4.2 Classification performance using time frequency information and independent frames

In figure 6.13 classification performance is given for the Bayes classifiers using a spectrogram per segmented seismic recording. The spectral frames were assumed independent of each other. Again the parzendc classifier performed best. Classification performance for the parzendc classifier was approximately equal compared to the previous experiment. But for all other classifiers, classification performance was worse compared to the previous experiment. Thus for most classifiers when using this experimental setup it is better to use a single averaged spectrum than to use a spectrogram and assume independent frames. Overall classification performance increases with classifier complexity. Except for the udc and qdc classifiers. The udc classifier outperforms the qdc classifier which is a little bit odd considering the number of frames per class (typically around 9000). In figure 6.14 a typical example of the feature representation is given for a window length of 256 and a window length of 2048. For the window length of 2048 the classes are slightly less overlapping at the top and bottom. Furthermore there is also a tail to the right of three of the four point clouds. These tails rotate the qdc distributions a little bit to the right preventing this classifier to take advantage of the less overlapping regions at the top and bottom of the point constellations. However the less flexible udc classifier can only place its distribution in the uncorrelated feature directions which is better in this case. The influence of the window length on classification performance is higher compared to the previous experiment. Classification performance is higher for the longer window lengths when using the parametric Bayes classifiers. The non parametric Bayes classifier performed better with shorter window lengths.

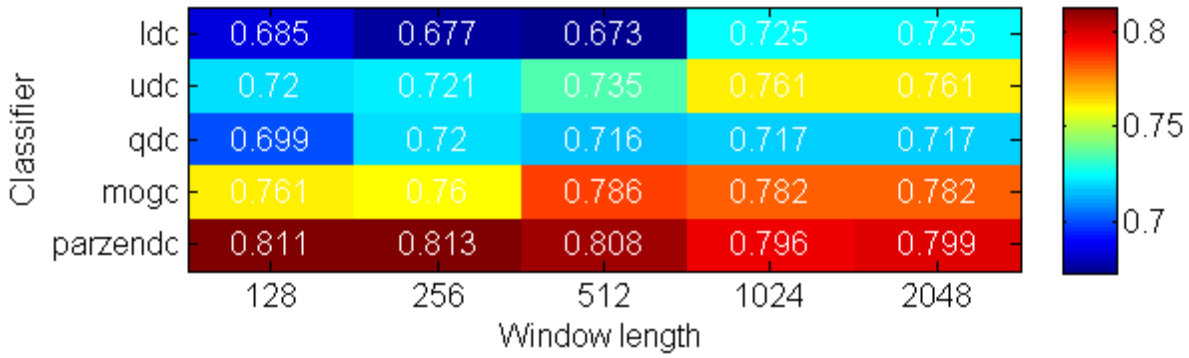

Figure 6.13: Classification performance using time frequency information and independent frames

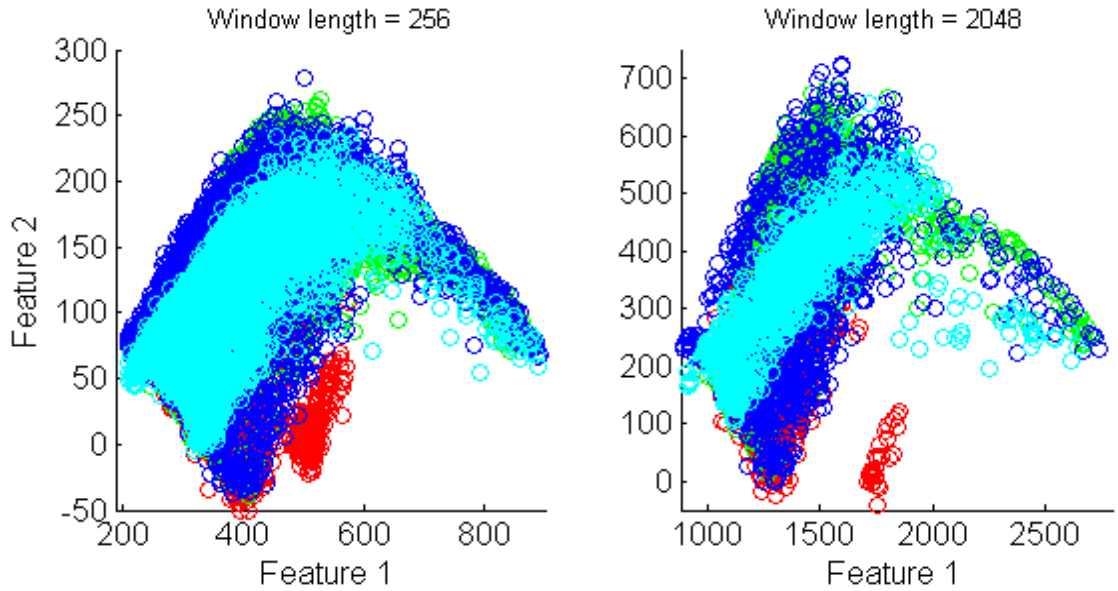

Figure 6.14: Typical examples of feature representations for this experiment using two different window lengths

6.4.3 Classification performance using time frequency information and hidden Markov models

In figure 6.15 classification performance is given for the normal mixture model and the continuous hidden Markov model as a function of the number of components/states. Both models receive an equal number of normal components. However in case of the normal mixture model the components are organized into one large normal mixture and the ordering of the frames within the sequence is of no importance. The continuous hidden Markov model uses states and each state receives one normal component. For this classifier the ordering of frames within a sequence of feature vectors is (or can be) of importance. Although both models receive the same number of normal components the hidden Markov model is the more complicated classifier because it also estimates a state prior and state transition matrix. In figure 6.15 one can see that the normal mixture classifier outperforms the continuous hidden Markov model almost consistently. This can be explained by the fact that the ordering of the feature vectors for this feature representation is not modelled by the hidden Markov model. Instead the hidden Markov model does approximately the same as the normal mixture classifier. The resulting feature representations of the seismic recordings are too far apart in the feature space for the hidden Markov model to take advantage of the state transition matrix (see figure 6.17). The resulting state transition matrices for this feature space are therefore almost diagonal. Thus the extra complexity of the hidden Markov model is a disadvantage for this feature representation. Classification performance for both the normal mixture classier and the continuous hidden Markov model seem to level out towards the higher number of components/states. Classification performance is slightly less compared to the results obtained using the parzendc classifier in the previous experiments. Classification performance is equal for the HMM and GMM model whilst using only one component/state. This is also what one would expect.

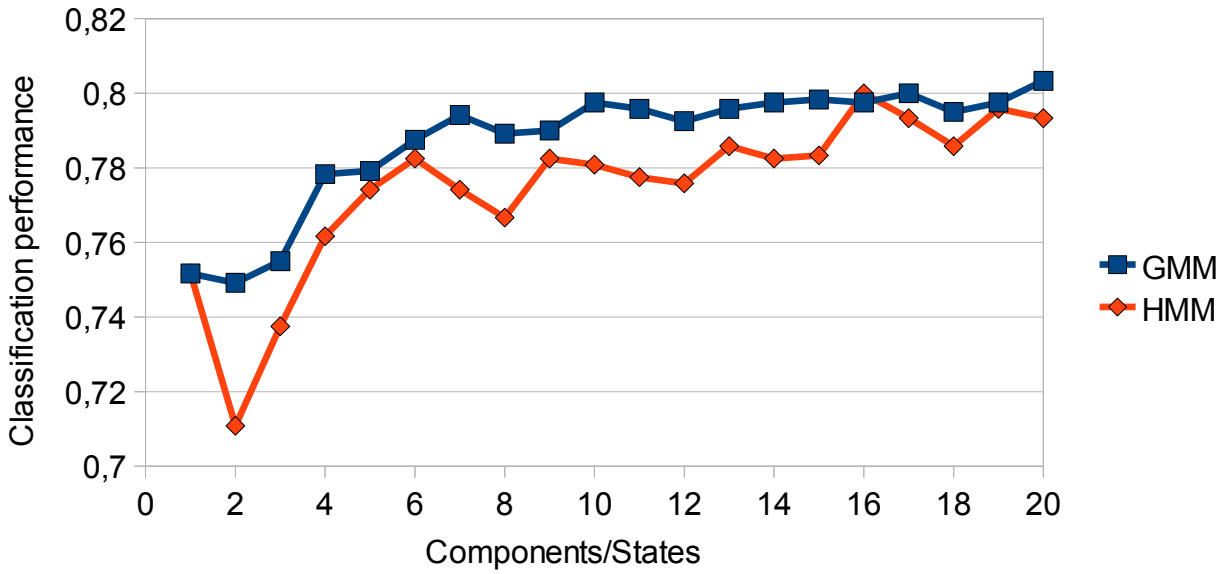

Figure 6.15: Classification performance normal mixture model (GMM) versus continuous hidden Markov model (HMM) in the original feature space (feature space used up until so far)

To bring the seismic recordings closer together in the feature space one can take the first derivative of each corresponding feature vector sequence. A typical segmented seismic recording results in 90 (four dimensional) feature vectors. Taking the first derivative of this sequence results in a new feature vector sequence of length 89. The dimensionality of the feature vectors remains the same. The new feature vectors contain the displacement between two consecutive feature vectors in the original feature space. Let us call this new feature space the delta feature space (see again figure 6.9).

In figure 6.16 classification performance is given for the normal mixture model and the continuous density hidden Markov model as a function of the number of components/states in the delta feature space. This time the hidden Markov model outperforms the normal mixture classifier. Typically the probability mass was also better divided inside the state transition matrices. The resulting state transition matrices were not almost diagonal any more. Classification performance for the hidden Markov model in the delta feature space is clearly better compared to the performance of the hidden Markov model in the original feature space. The normal mixture model clearly suffers from the new feature representation. A typical example of one of the classes in both feature spaces is given in figure 6.17. Finally a last experiment was performed using both the original and the delta feature vectors in a new set of combined feature vectors. Classification performance was worse whilst using these combined feature vectors (worse compared to the results achieved using only the delta feature space). Classification results of this experiment are not given.

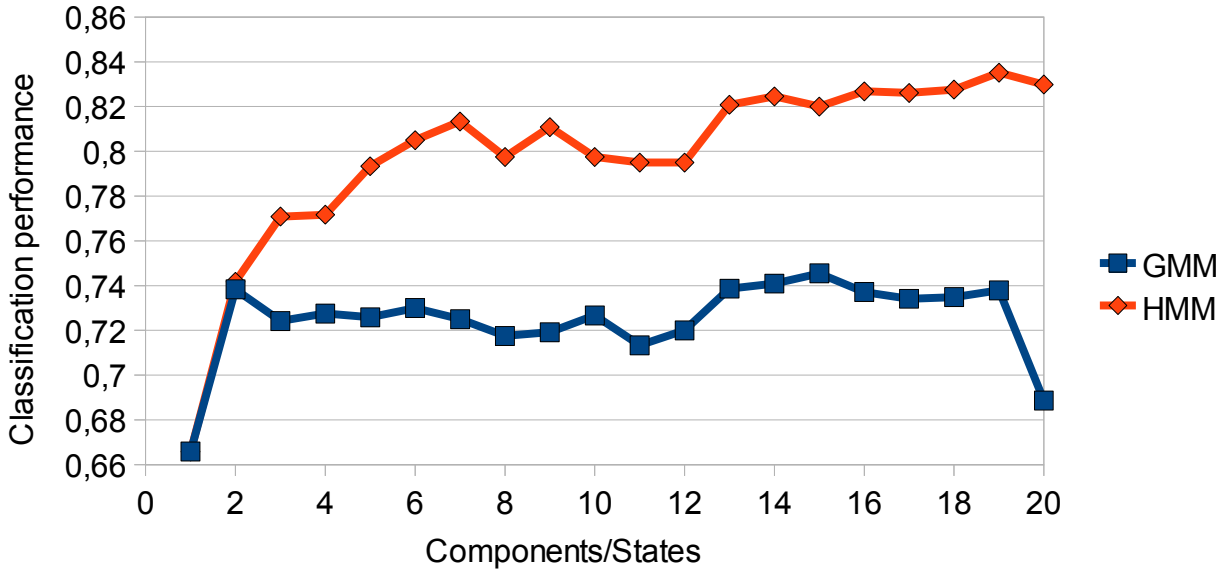

Figure 6.16: Classification performance normal mixture model versus continuous hidden Markov model in the delta feature space

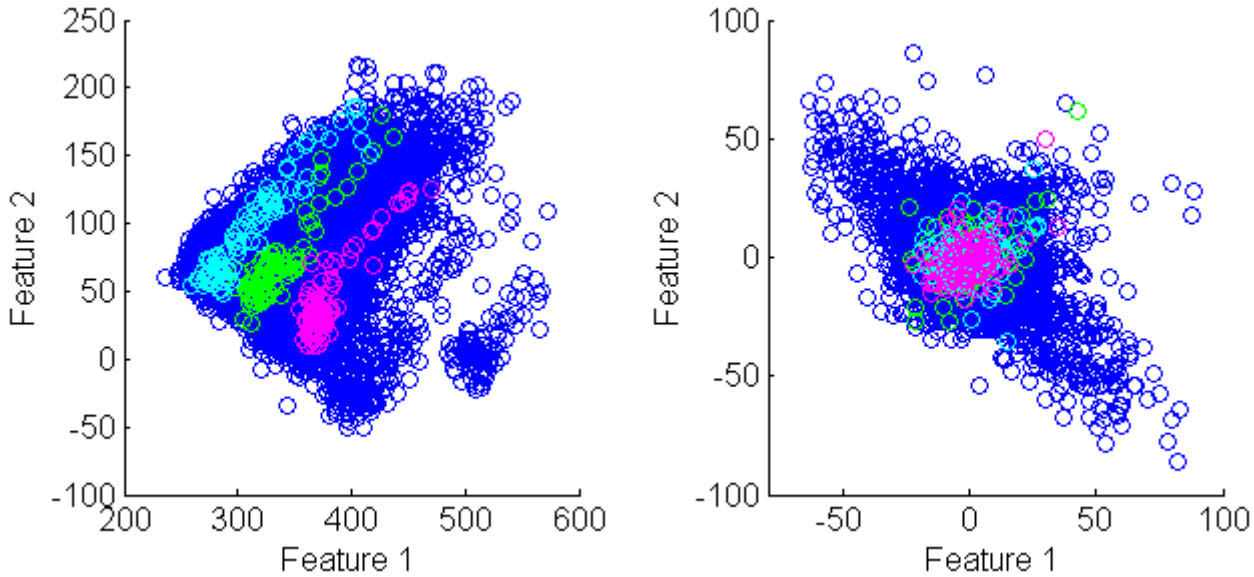

Figure 6.17: Original feature representation of one of the classes (blue circles). And three highlighted recordings (green cyan and magenta circles) (Left). Delta feature representation of one of the classes (blue circles). And three highlighted recordings (green cyan and magenta circles) (right)

In figure 6.18 classification performance is given for the discrete emission density hidden Markov model as a function of the number of states and the number of symbols or clusters. Classifier complexity of the discrete hidden Markov model is controlled by these two parameters. For this experiment again the delta feature space was used because it is the more interesting feature space for hidden Markov models. When using the discrete hidden Markov model a symbol is assigned to each feature vector. In this experiment the symbols are assigned by the k-means clustering algorithm. The assigned symbols are fixed and cannot change during training. This in contrast to the continuous hidden Markov model were the normal components can move and change shape during training. In figure 6.18 best classification performance is achieved using seven or eight symbols. The choice of the number of states seems to be less of importance. But classification performance is considerably less when using three or less states indicating that the discrete hidden Markov model is influenced by or can take advantage of the sequence ordering. The emission density functions for all classes are equal when using only one symbol. This means that in that situation the probability of occurrence on a given test recording only depends on the state prior and state transition matrix. As a result test recordings are always assigned to the same class. Therefore the classification results are exactly 0.25 for the symbol count of one. On our dataset discrete hidden Markov models performed worse compared to the continuous hidden Markov models.

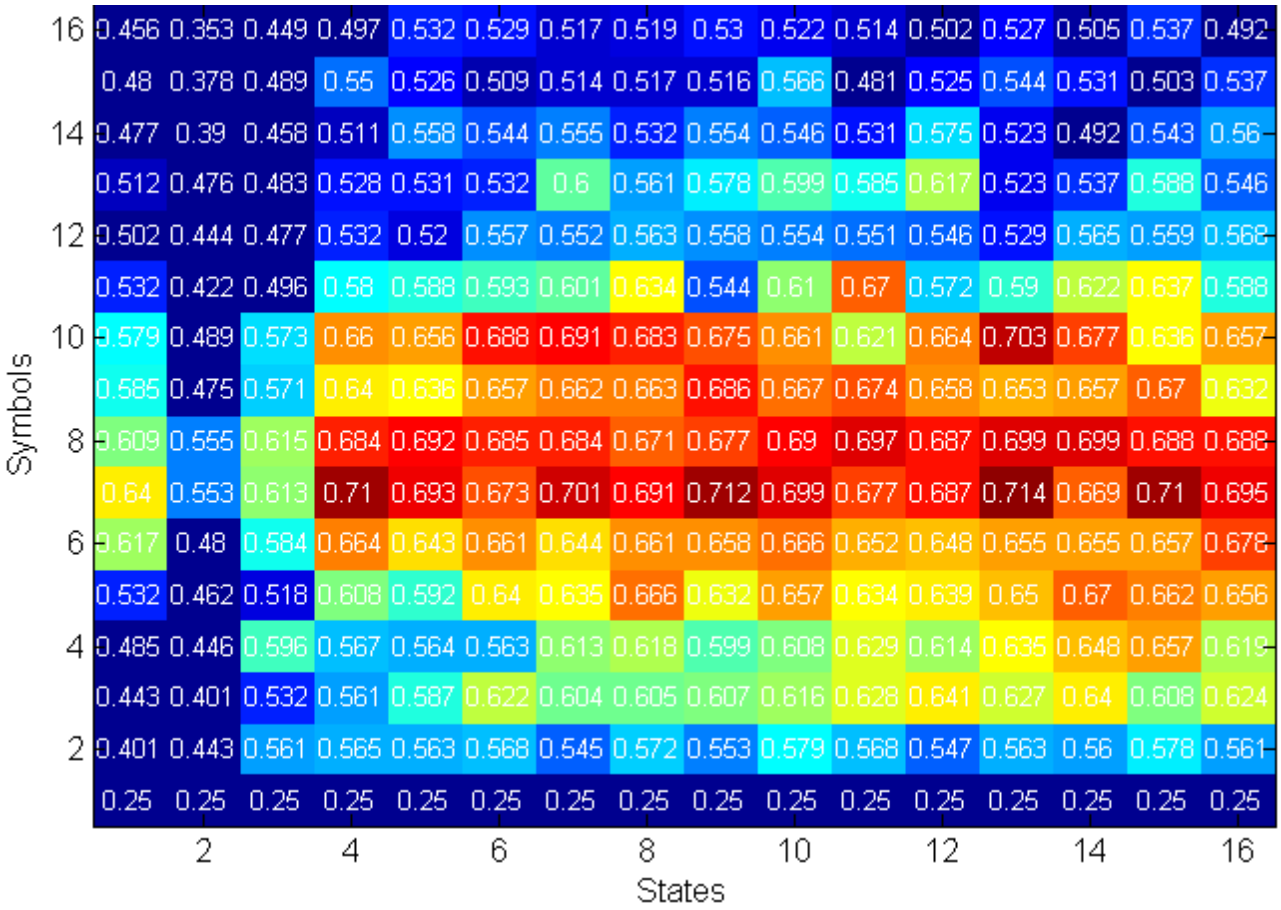

Figure 6.18: Classification performance of the discrete hidden Markov model in the delta feature space as a function of the number of states and the number of symbols

6.4.4 Classification performance using time frequency information and concatenated frames in the feature space

In figure 6.19 classification performance is given for several classifiers as a function of the number of averaged frames. The dimensionality of the resulting feature vectors is four times the number of frames (Again four DCT coefficients were used). Clearly classification performance is significantly better compared to the previous experiments. The ldc and fisherc classifiers performed best. But most other more complicated classifiers performed remarkably well considering the high dimensionality of the feature vectors. The parzenc classifier uses one smoothing parameter for all classes. It is possible that the distances between observations in one class vary too much from the distances between observations in the other classes. The different distances cannot all be modelled effectively using one smoothing parameter. If that is the case frame probabilities are computed using a very suboptimal smoothing parameter for each class resulting in numerical issues such as zero probabilities. This might explain the poor (random) classification performance for this classifier when using 33 frames and more. The quadrc classifier shows a very odd behaviour. First the classification performance decreases towards a minimum with an increasing number of frames. After this minimum the classification performance increases again with an even further increasing number of frames. That is not what one would expect. However there is a logical explanation for this behaviour in the internal construction of this classifier. When the dimensionality increases the covariance matrices become badly scaled and close to singular reducing classification performance. When the covariance matrices actually become singular the corresponding classifiers are replaced by fisher classifiers. When the dimensionality increases even further more and more classifiers are replaced by fisher classifiers. Therefore eventually classification performance is equal to the fisher classifier. It would have been better to use a regularization term for this classifier. Remarkably overall classification performance is not influenced much by the number of frames. The normal mixture Bayes classifier did not work and is not included in the experimental results.

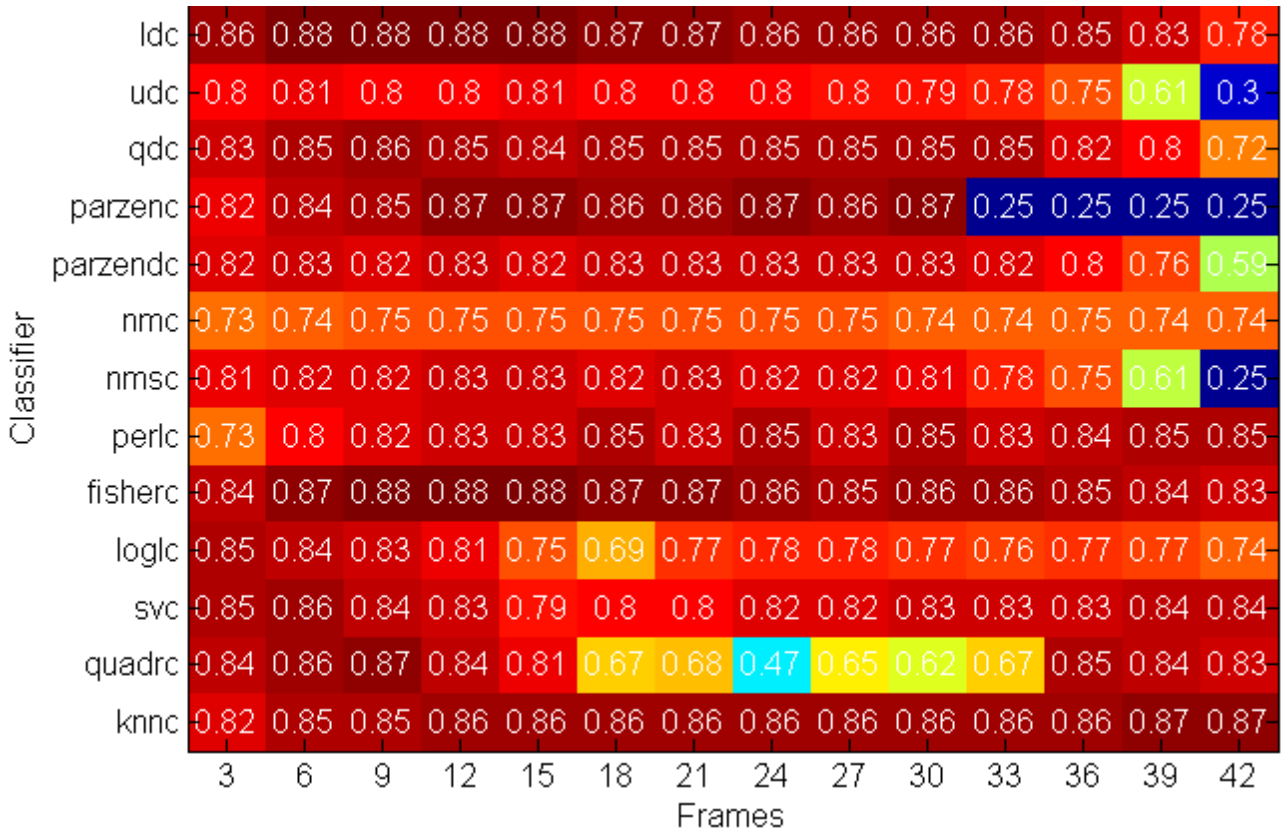

Figure 6.19: Classification performance using time frequency information and concatenated frames in the feature space

6.4.5 Classification performance using time frequency information and concatenated frames in the dissimilarity space

In figure 6.20 classification performance is given for several classifiers as a function of the number of concatenated frames in the dissimilarity space. The number of examples in the representation set used for the results in figure 6.20was 120 (30 examples for each class). Several experiments with different numbers of representation examples were performed (see figure 6.21). But the representation set of 120 examples both gave a interesting result for most of the classifiers and also achieved a classification performance of 91% for the ldc and fisherc classifiers. A classification performance of 91% was also achieved for the fisherc classifier whilst using representation sets of 200, 240, 280 and 320 examples but for these representation sets more classifiers produced a random classification performance. For the results in figure 6.21 the number of frames was fixed and set to 42. In figure 6.21 most of the Bayes classifiers clearly suffer from a higher number of representation examples. But the qdc classifier did remarkably well for the higher number of representation examples when compared for example with the udc and ldc classifiers. The discriminant classifiers were far less sensitive to a changing number of examples in the representation set. Except for the scaled nearest mean classifier. In figure 6.20 the dimensionality and the number of objects per class do not change as a function of the number of frames. Only the shape of the classes and the distances between the recordings change. The udc, parzenc and nmsc classifiers are clearly sensitive and negatively influenced by the changing class shapes and increasing distances.

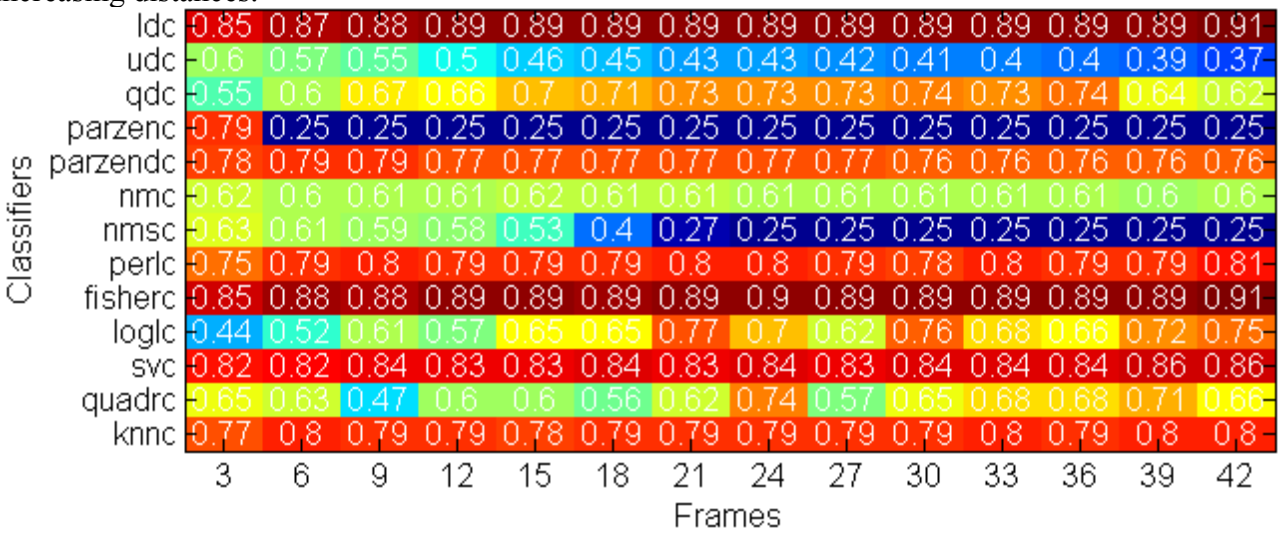

Figure 6.20: Classification performance using time frequency information and concatenated frames in the euclidean dissimilarity space (representation set size = 120 examples)

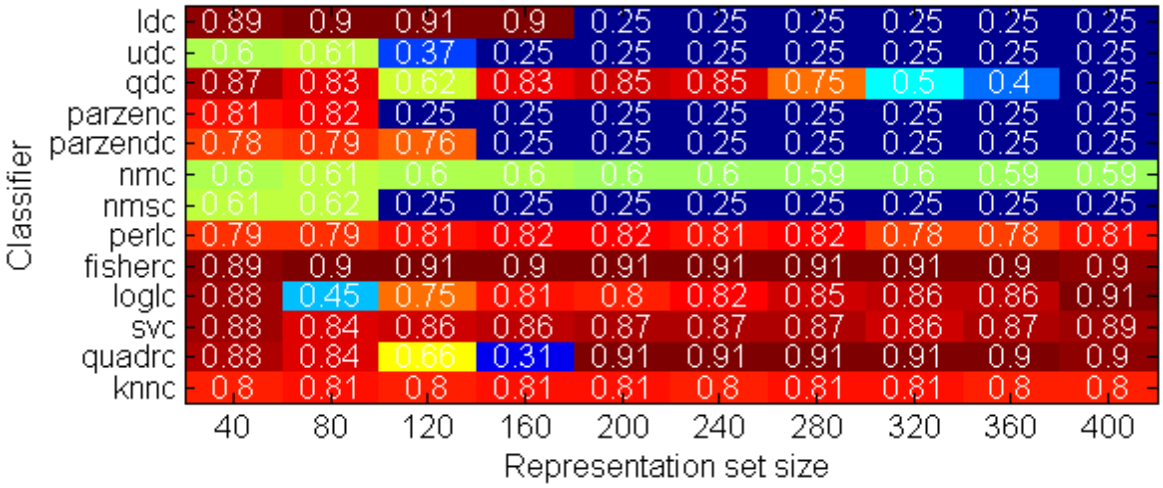

Figure 6.21: Classification performance using time frequency information and concatenated frames in the euclidean dissimilarity space (number of frames = 42)

In figure 6.22 the final classification results are given for this chapter. The experiment was performed in almost the same conditions compared to the previous experiment. Except for this experiment modified Hausdorff distances were used. The number of examples in the representation set used for this experiment was 80. Again several experiments with different numbers of representation examples were performed. But best results were achieved using 80 representation examples. Clearly classification results are not as good as the results obtained using euclidean dissimilarities. The qdc classifier performed best followed by the ldc and fisher classifiers. Looking at the results given in figure 6.20 and figure 6.22 one might conclude that the ordering of the spectral frames inside the concatenated feature vectors is of importance. The modified Hausdorff distance allows seismic recordings that are similar but not exactly aligned in time to still have a small resulting distance to each other. Both Within class and between class recordings receive a distance that is equal or smaller compared to the euclidean distance. Thus typically the distances between recordings are smaller, this might also be the cause of the reduced classification performance (similar recordings of different classes also become closer to each other). Furthermore one might also conclude that the alignment of the segmented seismic recordings is very good. Otherwise one would not find such good classification results whilst using such a high number of frames.

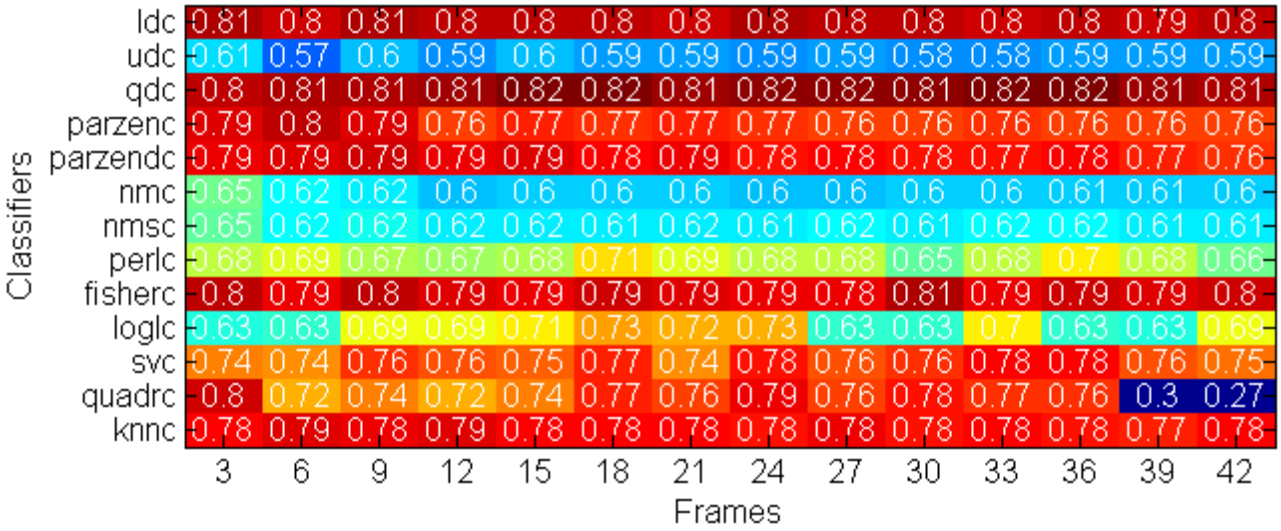

Figure 6.22: Classification performance using time frequency information and concatenated frames in the modified Hausdorff dissimilarity space (representation set size = 80)

6.5 Observations and conclusions

From the results one can observe and conclude the following:

- For most of the classifiers used in the experiments it is better to use a single averaged spectrum (only frequency information) than to use a spectrogram (time frequency information) and combine the spectral frame probabilities using the product rule. (similar conclusion compared to the conclusion in chapter 5 but now whilst using several different classifiers).
- In the original feature representation developed in chapter 5 hidden Markov models were unable to model the sequence ordering because individual recordings were too far apart from each other in the feature space. Individual recordings were modeled by the normal components instead. Similar to what the normal mixture model does. In the delta feature space the continuous hidden Markov model performed better and was able to model the sequence ordering better. Classification performance improved over the single averaged spectrum representation.
- The discrete hidden Markov model performed worse compared to the continuous hidden Markov models but was also able to model the sequence ordering in the delta feature space.
- Incorporating both time and frequency information by concatenating several averaged frames in one long feature vector improves classification performance significantly over the single averaged spectrum representation, both in the feature as well as in the dissimilarity space.
- The ordering of the averaged frames inside the resulting feature vectors is of importance whilst using the last two classification strategies.
- The alignment of the segmented seismic recordings was very good. Thus averaged frames resulting from different segmented seismic recordings at least often correspond to the same underlying physical phenomenon (or at least have similar spectral characteristics).

For this study a couple of experiments were also performed using maximum entropy models[23] [24][25]. The principle of maximum entropy states that if incomplete information about a probability distribution is available, the only unbiased assumption that one can make is a distribution that is as uniform as possible under the constraints of the available training material. Maximum entropy models model the class posterior probability density function directly. This in contrast to the maximum likelihood models such as the Bayes classifiers which model the class conditional density functions. The training material is incorporated in the class posterior probability density function via the use of features. Typically features are defined as binary valued functions which both depend on the observation and on the class variable. When binary features are used the optimization surface is convex in terms of the model parameters. Thus a global optimum can be found (using for example the generalized iterative scaling algorithm) which is very interesting (The main reason why I started investigating these models). Maximum entropy models are very popular in natural language processing applications because the training material of these applications can be expressed conveniently in terms of binary feature functions. However for our volcano data set the use of binary valued features is unnatural and difficult. The main difficulty is to find a good set of binary features (how many binary features to use?, Where to place the binary features in the original continuous valued feature space? And what size should each binary feature occupy in the original continuous feature space) Optimizing a large number of binary features is very computational expensive (Convergence towards a global optimum for a large model can be slow). Furthermore the maximum entropy distribution is uniform outside the region occupied by the binary features potentially resulting in poor generalization. The maximum entropy model is simply not practical for our problem.

7 Multiple stations

7.1 Introduction

Up until so far the segmented seismic recordings from one station were used to find a good pattern classifier (feature extraction block and classification block). However the data set received for this study also contains segmented seismic recordings from several other seismic stations.

It is of interest to see if the pattern classifier developed in the previous two chapters also gives comparable classification results on the segmented seismic recordings of the other stations. How well does the chosen block, block parameter and classifier combination generalizes to other stations? And can one successfully train a classifier on the seismic recordings of one station and find similar classification results when testing performance on the seismic recordings of the other stations? Is the same seismic event registered in a comparable way at different locations by different seismometers?

Furthermore it is of interest to see if classification performance can be improved by combining classification results of several stations.

7.2 Experimental setup

7.2.1 Testing criteria

Again the hold out estimate was used. The holdout estimate was repeated ten times each with a different random permutation of the volcano dataset. The same random seed was used that was also used earlier for all the experiments in chapter 5 and 6. Mutually exclusive random index permutations were used for selecting the training and test sets even when the training set station was different from the test set station. Otherwise the resulting performance measure might be positively biased. Remember that a given recording i corresponds to the same seismic event for all selected stations and that a given seismic event might look very similar when registered by different seismometers at different locations. When selecting training and test sets from multiple stations in the classifier combining experiment the same random index permutations were used for all the stations.

7.2.2 Data set

For the following experiments recordings from all five a priory selected stations were used. The first 133 seismic recordings from each station and each class were selected. A total of 2660 seismic recordings were used for the experiments (5*4*133=2660). Again all classifiers were trained using 100 randomly chosen recordings per class. Classification performance was tested on the remaining 33 recordings of each class.

7.2.3 Classifier

For these experiments the ldc classifier was used. The ldc classifier was trained on the euclidean dissimilarity feature vectors that were described/found in chapter 6. The number of averaged frames of each recording was set to 42. The number of representation examples used to form the dissimilarity space was set to 120. These settings gave the best classification results for the ldc classifier in chapter 6. The ldc classifier was used because it gave superior classification performances whilst using the datasets originating from the other stations. The experimental results of the other less performing classifiers are not included in this work.

7.2.4 Combining rules

In the classifier combining experiment the recordings from different seismic stations were combined at the feature level[3]. An ldc classifier was trained on the recordings of each station involved in each given combination. Obviously the sensor or station classifiers involved in the combinations were all trained in a different feature space. Each station classifier produces a set of class posterior probabilities $p(\omega_j|\boldsymbol{x})$ when presented with a test object. The sensor or station class posterior probabilities were combined using a combination rule, which is itself a classifier defined on a feature space of posterior probabilities. The combining rule provides an estimate of (or a value that is proportional to) the class posterior probability conditioned on the recordings of all involved stations $p(\omega_j|\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$. The classifier combining architecture is given in figure 7.1. The obvious question for this architecture is: Given the outputs of the sensor or station classifiers what is a good choice for the combination rule?

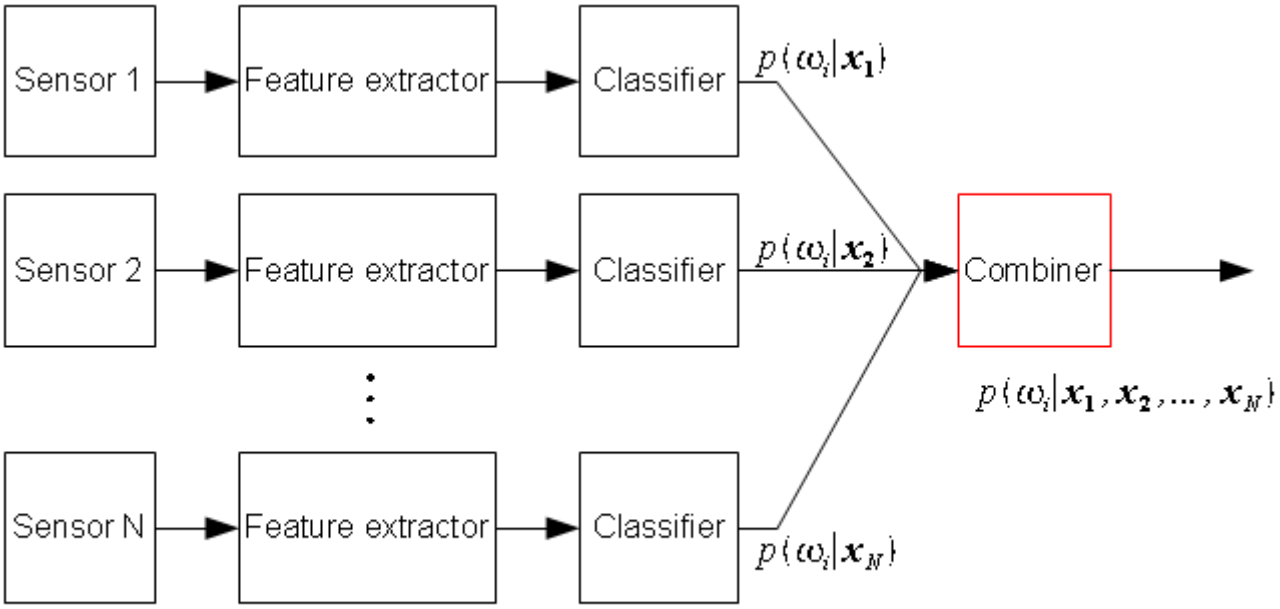

Figure 7.1 Classifier combining architecture

Four combination rules were used in the combining experiment:

Product rule

The product rule assumes conditional independence of the recordings originating from different stations. The recordings $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ are assumed conditionally independent given the class. In all the experiments in this study equal class priors were used (class priors are ignored in the class posterior probability computations). Whilst using the product rule the test object is assigned to the class for which the product over the class posterior probabilities is largest The product index is running over the stations (See also equation 7.1). The product rule is usually applied when each classifier receives input from different (independent) sources.

$$\prod_{i=1}^{N} p(\omega_j|\boldsymbol{x}_i) > \prod_{i=1}^{N} p(\omega_k|\boldsymbol{x}_i) \quad k = 1, \ldots, C; k \neq j$$

Equation 7.1: Product rule for equal class priors according to [3]

Sum rule:
The sum rule is sometimes also referred to as Bayes voting. Intuitively this combining rule is similar to the majority vote combining rule. The majority vote combining rule is a combining method at the decision level. The majority vote combining rule assigns one to the class for which the corresponding posterior probability is largest and zero to all other classes. The 'vote' is assigned to one class. In contrast the sum combining rule divides the 'vote' in fractions to the different classes. The test object is assigned to the class for which the sum over the fractional 'votes' is largest. See equation 7.2. Mathematically one assumes that the posterior probabilities $p(\omega_j|\boldsymbol{x})$ are similar to the class priors $p(\omega_j)$ which is unrealistic in many cases. However the combiner rule is robust and often used for combining classifiers that were trained on common input patterns. Which is perhaps also the case for our data set (Possibly the recordings of different stations corresponding to the same seismic event are very similar).

$$\sum_{i=1}^{N} p(\omega_j|\boldsymbol{x}_i) > \sum_{i=1}^{N} p(\omega_k|\boldsymbol{x}_i) \quad k=1,\ldots,C\,; k \neq j$$

Equation 7.2: Sum combining rule for equal class priors according to [3]

Max rule:
The max combining rule is a simplification/approximation of the sum rule. The assumption for the max combining rule is that there is one classifier output for each class that clearly dominates or is much larger compared to the other classifier outputs. The dominating output for a particular class is sometimes also referred to as the experts decision. Instead of summing over all the class posterior probabilities involved in the combination the max rule only uses the expert class posterior probabilities that would have otherwise dominated the outcome anyway. The max combining rule for equal class priors is given in equation 7.3.

$$\max_i p(\omega_j|\boldsymbol{x}_i) > \max_i p(\omega_k|\boldsymbol{x}_i) \quad k=1,\ldots,C\,; k \neq j$$

Equation 7.3: Max combining rule for equal class priors according to [3]

Trained combining rule:
Another alternative is to use the class posterior probabilities resulting from the sensor/station classifiers as the input or features to a trained combining classifier. In the experiment several different classifiers (all classifiers of chapter 6 except the hidden Markov classifiers) were trained on the output posteriors of the station classifiers.

7.3 Results

7.3.1 Generalization to other seismic stations

In figure 7.2 the classification results are given for the five a priory selected seismic stations. On the vertical axis the training set station is given. And on the horizontal axis the test set station is given. Classification results for the diagonal elements were achieved by both training and testing on the recordings originating from one given seismic station. Classification results for the off-diagonal elements were achieved by training on the recordings of one seismic station and testing on the recordings of another station. The diagonal classification results are all within 3% of each other indicating that the chosen pattern classifier is robust. The chosen block, block parameter and classifier combination is also a good solution for the recordings originating from the other stations. Like one would expect the off-diagonal classification results are almost always worse compared to the diagonal classification results. This might be caused by varying ground and instrument conditions at different locations. But the classification results of the off-diagonal elements are much better compared to results achieved using the random classifier indicating that seismic events are registered in a similar way by different seismometers at different locations. One would expect a somewhat symmetric classification result (symmetric around the diagonal). For example when the pattern classifier is trained using the recordings from station i and classification performance is measured using the recordings from station j one would expect similar classification results when station i and station j exchange place. However for the experimental results given in figure 7.2 this is at least not always the case. Especially when the ref station is involved in training or testing.

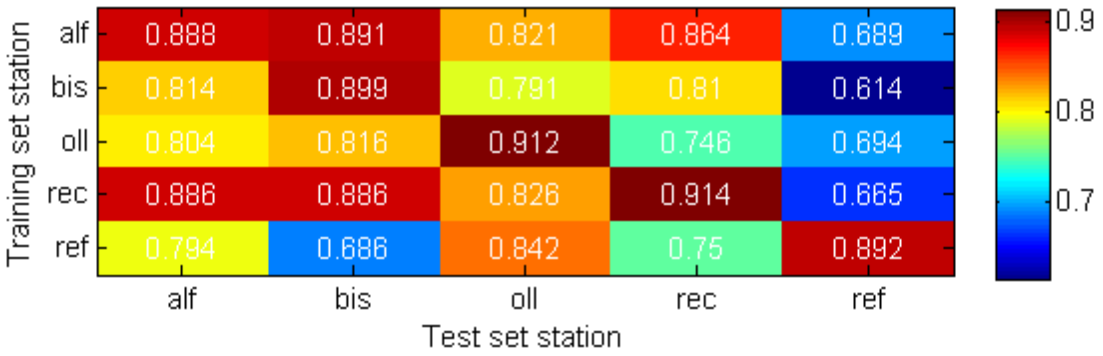

Figure 7.2: Classification performance for all five a priory selected stations using the ldc classifier in the euclidean dissimilarity space

7.3.2 Combining seismic stations

In figure 7.3 the classification results are given for the seismic station combining experiment. On the horizontal axis the seismic station combining method is given. On the vertical axis the seismic station combination permutation is given. Four classifier combining methods were used in this experiment. The product, sum and max combining rules were used in combination with all possible combinations of the five selected stations. When combining two or three stations there are ten possible combinations, when combining four stations there are five possible combinations and finally when combining all five stations there is only one possible combination (the ordering of stations is of no importance for these combining rules). When combining two stations the first combination permutation index corresponds to the combination of the alf and bis station. The second index corresponds to the combination of the alf and oll station etc... Best classification performance was achieved with the combination of the alf, oll and rec stations whilst using the max combining rule. The oll and rec stations are the individually best performing stations thus this results does not come completely as a surprise. In this experiment the combined classification result was always better compared to the individual best performing station involved in the combination. In the experimental results given in figure 7.3, the choice of the stations inside a combination is of more importance than the choice of the combining rule.

A couple of experiments were performed using several different classifiers in the classifier output space. For these experiments all five stations were used. The dimensionality of the classifier output space is the product of the number of stations and the number of classes. Thus the dimensionality of the resulting output space was 20. Best classification performance was achieved using the parzenc classifier. The classification result of the parzenc classifier in the classifier output space was worse compared to the untrained combining rules in the same output space. Indicating that the trained classifiers used for this experiment were either over trained or could not find a good generalization (the first cause is more likely in case of the parzenc classifier).

The classification performance variance as a function of the combining permutation decreases with an increasing number of stations involved in the combination. Of-course this is also what one would expect. The classification improvement step is largest whilst going from the individual performances to a combination of two stations.

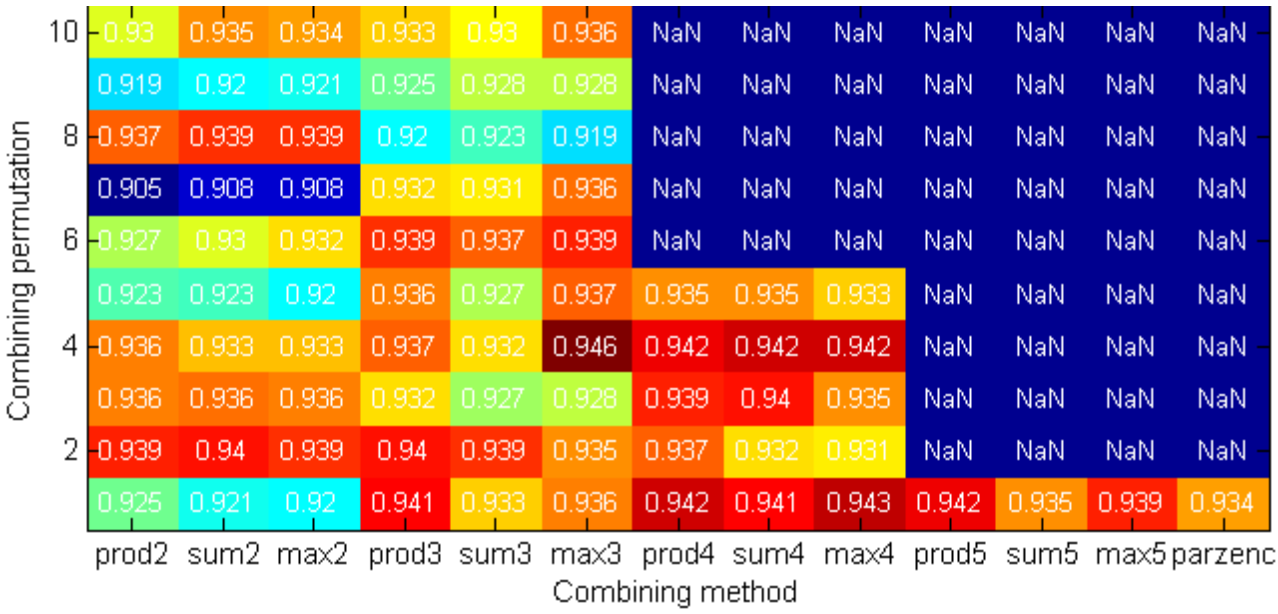

Figure 7.3: Classification performance for several combinations of the seismic stations

7.4 Observations and conclusions

From the results one can observe and conclude the following:

- One can conclude that the recordings from different stations are informatively different. Otherwise combining the recordings from different stations would not have improved classification performance. But one can also conclude that the informative differences are small. Only a maximum improvement of 3%-4% was achieved whilst combining seismic stations. The differences in recording representation between stations are much larger. Otherwise the off-diagonal classification performance would have been better.
- Best individual classification performances were achieved when both the training and test sets originate from the same station. Individual classification results of the five seismic stations were very similar indicating that the (untrained) pattern classifier is also a good solution for the recordings of the other stations.
- Classification results were almost always significantly less when training was performed on the recordings of one station and testing was done on the recordings of another station. Thus preferably one pattern classifier is trained on the recordings of each station. It would have been interesting to see the classification results of a single pattern classifier that was trained on the recordings originating from all seismic stations. One would suspect that these results would have been somewhere in between the off-diagonal and diagonal results.
- All combining rules used in this experiment improved classification performance over the best individual performing station involved in the combination (regardless of the station combination). The choice of the combining rule was not critical. The choice of the stations involved in the combination was of more importance.
- Training a combining classifier in the classifier output space did also improve the classification performance over the best individual performing station. The choice of the classifier was critical. Only the parzenc and the nmsc classifier performed better than the best individual performing station. The trained combiner performed worse compared to the untrained combining rules.

8 Discussion and Conclusions

The Nevado del Ruiz volcano is an active and dangerous volcano in the Andean volcanic belt. Measuring seismic activity is one of the most reliable and widely used techniques to monitor and predict renewed volcanic activity. Regions of interest are still classified by hand by the VSOM (Volcanological and Seismological Observatory Manizales) staff. In this study a pattern classifier was described/developed capable of discriminating reliably (±90%) between four frequently occurring seismic event types. (Frequently occurring in the received data set for this study) The received data set for this study also contained four additional event types but these event types were not included because these event types were poorly sampled in the received data set.

Frequency information in the form of a single spectrum per segmented seismic recording is often used by the VSOM staff to discriminate between different event types. Frequency information is also often successfully used in other studies involving seismic signals[13][14][15]. Therefore frequency information was also used in this study. Two popular types of spectral estimation methods were tested in combination with three dimensionality reduction methods. Overall the choice of the block and block parameters was not critical but by careful tweaking classification performance was improved by several percentages over the classification results achieved by the average block and block parameter choice (with a couple of exceptions of-course). Perhaps a careful study using the wavelet transform might improve classification performance even further[9].

The chosen block and block parameters in chapter 5 were used in combination with five different classification strategies and a large number of classifiers. Both Bayes and discriminant classifiers were used in the experiments. Using both time and frequency information improved classification performance over the single spectrogram representation. But only when the ordering of frames was taken into account in the representation. The continuous hidden Markov model improved classification results over the single spectrum representation whereas the naive Bayes classifiers trained on the same time frequency representation did not improve classification results. The concatenated spectrogram representation also improved classification results significantly over the single spectrum representation (Both in the feature space as well as the euclidean dissimilarity space). Classification performance was not improved whilst using the concatenated spectrogram representation in the Hausdorff dissimilarity space. The ordering of the frames in the spectrogram representation is of no importance whilst using the Hausdorff distance. Classification performance using the dissimilarity representation might be improved further using other distance measures. The classification performance of the hidden Markov models might be improved by the spectral averaging operation also done in the experiments with the concatenated feature vectors.

In chapter 7 we looked at the recordings of multiple stations. The untrained pattern classifier developed in chapter 5 and 6 also generalizes to other stations. The untrained pattern classifier might even also be a good solution for the automatic classification of seismic signals at other volcanoes. Generalization of a trained pattern classifier to other stations was almost always (much) worse. In this study we looked at combining the recordings resulting from different stations at the feature level[3]. At the feature level, classification performance improvement were achieved of approximately 3-4% over the individual best performing stations. It is of interest to see if classification can be improved even further by combining classifiers at the data level.

The received data set for this study was relatively small. Which was great for testing. All recordings in the received data set originate from a small period in time. Resulting recordings from a given

class of events might change over time due to changing ground path and instrument conditions. It is of interest to see how well the developed pattern classifier generalizes when presented with a larger data set.

References

Text books on geology and volcanic seimology

[1] V.M. Zobin, “Introduction to volcanic Seismology” Elsevier
[2] D.J. Doeglas, G.B. Engelen, G.C. Maarleveld, A.J. Pannekoek... “Algemene geologie” 1976

Text book on pattern recognition

[3] Andrew and Webb, “Statistical pattern recognition (second edition)” Wiley

Papers on pattern recognition

[4] Sheetal Lahabar, Pinky Agrawal, P.J. Narayanan “High performance pattern recognition on GPU”
[5] R.W.M. Keunen, R. Hoogenboezem, R. Wijnands, A.C.M. Van den Hengel, R.G.A. Ackerstaff, “Introduction of an embolus detection system based on analysis of the transcranial doppler audio signal” informa, Journal of medical engineering and technology

Text books on signal processing

[6] James H. McClellan, Ronald W. Schafer and Mark A. Yoder, “Signal processing first” Pearson Prentice hall

[7]A.W.M. Van den Enden en N.A.M. Verhoeckx, “Digitale signaalbewerking” Delta press

[8] Khalid Sayood, “Introduction to data compression” Wiley

Papers on seismic signal classification

[9] Mohammed Bendrahim, Adil Daoudi, Khalid Benjelloun, “Discrimination of seismic signals using artificial neural networks” World academy of Science and technology 4 2005

[10] Robert P.W. Duin, Mauricio Orozco-Alzate, John Makario Londono-Bonilla, “Classification of volcano events observed by multiple seismic stations” 2010

[11] J.C. Lahr, B.A. Chouet, C.D. Stephens, J.A. Power, R.A. Page, “Earthquake classification, location, and error analysis in a volcanic environment: implications for the magmatic system of the 1989-1990 eruptions at redoubt volcano, Alaska” Journal of volcanology and geothermal research 62 (1994) 137-151

[12] Mauricio Orozco-Alzate, Marina Skurichina, Robert P.W. Duin, “Spectral characterization of Volcanic earthquakes at Nevado del Ruiz volcano using spectral band selection/extraction techniques”

[13] S. Scarpetta, F. Giudicepietro, E.C. Ezin, S. Petrosino, E. Del Pezzo, M. Martini, and M. Marinaro, “Automatic classification of seismic signals at Mt. Vesuvius volcano, Italy, using neural networks” Bulletin of the Seismological society of America.

[14] Mauricio Orozco, Marcelo E. Garcia, Robert P.W. Duin, Cesar G. Castellanos, “Dissimilarity-based classification of seismic signals at the Nevado del Ruiz volcano” Earth

sciences research journal

[15] M.C. Benitez, Javier Ramirez, Jose C. Segura, Jesus M. Ibanez, Javier Almendros, Araceli Garcia-Yeguas, Guillermo Cortes, “Continuous HMM-based seismic-event classification at Deception island, Antartica” IEEE transactions on geoscience and remote sensing vol. 45, No. 1, January 2007

[16] L. Galli, C. Castellani, G. Pace G. Saccorotti “Wavelet decomposition and advanced denoising techniques for analysis and classification of seismic signals”

Papers on expectation maximization

[17] Sean Borman,”The expectation maximization algorithm a short tutorial”

Papers on hidden Markov models

[18] Lawrence R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition” proceedings of the IEEE, VOL. 77 NO. 2 february 1989

[19] Jeff Bilmes, “What HMMs can do” UWEE technical report january 2002

[20] Manuele Bicego, Vittorio Murino, Mario A.T. Figueiredo, “A sequential pruning strategy for the selection of states in hidden Markov models” Pattern recognition letters 2003

[21] Manuele Bicego, Vittorio Murino, Mario A.T. Figueiredo, “Similarity-based classification of sequences using hidden Markov models” Pattern recognition 2004

[22] Tobias P. Mann, “Numerical stable hidden Markov model implementation” ,

Papers on maximum entropy models

[23] Adam L. Berger, Stephen A. Della Pietra, Vincent J. Della Pietra, “A maximum entropy approach to natural language processing” Association for computational linguistics

[24] Roman klinger, Katrin Tomanek, “Classical probabilistic models and conditional random fields” Algorithm engineering report

[25] Hanna M. Wallach, “Conditional Random Fields: An introduction” CIS Technical report