UNIVERSITY OF TARTU

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Institute of Computer Science

Field of Computer Science

Jürgen Jänes

Detection of emission line stars from the Gaia space telescope

Bachelor's Thesis

Supervisors: Sven Laur, PhD
Indrek Kolka, PhD

Author: .... "....." June 2009

Supervisor: .... "....." June 2009

Approved for defense

Professor: .... "....." June 2009

TARTU 2009

Contents

Introduction **4**

1 Astronomical Background **5**
- 1.1 Introduction 5
- 1.2 Gaia space telescope 7
- 1.3 BP/RP Photometer 8
  - 1.3.1 Effective line width 9
  - 1.3.2 Apparent magnitudes 10
- 1.4 Emission line stars 10
  - 1.4.1 Be stars 10
  - 1.4.2 Wolf-Rayet stars 11

2 Computer Science Background **12**
- 2.1 Support Vector Classification 12
- 2.2 Support Vector Regression 16
- 2.3 Principal Component Analysis 16

3 Materials and Methods **17**
- 3.1 Datasets 17
- 3.2 Classification 17
  - 3.2.1 Detecting H-$\boldsymbol{\alpha}$ emission 19
  - 3.2.2 Separating Be stars and Wolf-Rayet stars 20
- 3.3 Synthetic spectra 20
- 3.4 Effective width of a fixed spectral line 23

4 Results and Discussion **24**
- 4.1 Classification 24
  - 4.1.1 Detecting H-$\boldsymbol{\alpha}$ emission 24
  - 4.1.2 Separating Be stars and Wolf-Rayet stars 25
- 4.2 Line-width estimation 26
- 4.3 Discussion 27

5 Conclusion **30**

A Resümee **31**

References **32**

Introduction

It is a well-known fact of computer science that the density of transistors that can be fitted on an integrated circuit has been growing at an exponential rate, thus leading to significant advances in computational capabilities.

Moore's law is just one example of this kind of miniaturization. Other interesting examples are the number of megapixels in a consumer digital camera and, more on the topic of this thesis, the number of CCD elements used in astronomical cameras. Combined with other technological advancements, this has led to an exponential growth in astronomical observational data, which has led to an increase in using computer algorithms for analysing observational data.

The aim of this thesis is to conduct preliminary experiments on detecting and characterizing emission line stars from the low-resolution BP/RP photometer instrument of the Gaia space telescope. Gaia is an ambitious science mission of the European Space Agency, that is currently being built and is planned for launch in 2011.

Although we are solving an astronomy problem, we are mostly using standard machine learning algorithms to do this, which makes this topic suitable for defending a degree in computer science.

The main contribution of this thesis is the evaluation of Support Vector Machine algorithms for detecting and characterizing emission line stars from Gaia BP/RP spectra. First, we use Support Vector Machines in classifying emission line stars. Second, we validate Support Vector Regression for inferring the effective line width of a fixed spectral line. Due to scarce available data on emission line stars, we construct a naive model of emission line spectra to generate sufficient data for the effective line width estimation algorithm.

The main text of the thesis consists of four chapters. In the first chapter, we give the necessary astronomical background information on the problem we are solving. The second chapter is an overview of the computer science algorithms we use in our experiments. In the third chapter, we describe our datasets, a spectral model we used to generate data for the line width estimation experiments, as well as our experimental set-up. In the fourth chapter, we present and discuss our results.

1 Astronomical Background

This chapter is divided into four parts. In Section 1.1 we give a short introduction on the present state of high-throughput astronomical observations. Section 1.2 gives an overview of the Gaia space telescope. Section 1.3 describes the BP/RP Photometer of the Gaia telescope and Section 1.4 gives a short introduction to emission line stars.

1.1 Introduction

A human being can observe a few thousand stars in a clear night sky with the naked eye. All of these stars belong to the Milky Way galaxy, which is a disk-like structure consisting of approximately 200 billion stars. Under good viewing conditions (low light pollution, non-intoxicated observer), one can also see a few other galaxies, although the number of galaxies observable using telescopes is significantly higher. Current estimates on the total number of galaxies in the Universe are in the same order of magnitude as the number of stars in the Milky Way galaxy.

In addition to regular Sun-like stars and galaxies, various other objects have been observed in the night sky, including planets, dwarf planets, comets, asteroids, nebulae, white dwarfs, neutron stars, black holes and quasars. In an intriguing development, evidence of extrasolar planets (planets that orbit other stars besides the Sun) have been obtained in the last two decades.

The vast majority of this information is obtained by measuring the energy distribution of electromagnetic waves coming from these objects. Different instruments and techniques are being used to observe waves with different wavelengths (these are roughly divided into radio, infrared, optical, ultraviolet, X-ray and gamma ray). In addition to studying spectra, precise positional measurements of astronomical objects are often performed. Potential uses for these include studying the movement of stars in the Milky Way or projecting the trajectory of a near-Earth asteroid. Other, more exotic methods for studying celestial objects include neutrino detectors and gravitational wave experiments.

Traditionally, astronomical observations have been done manually, by selecting a set of interesting objects and observing them one object at a time. Recent advancements are making it possible to scan large areas of the sky simultaneously, thus significantly increasing the amount of data requiring analysis [Ree09]. Examples of such a project are the Sloan Digital Sky Survey (SDSS), which has covered a quarter of the selestial sphere using a 2.5-meter dedicated telescope located in New Mexico, thereby obtaining measurements of 100 million galaxies and the Hipparcos satellite which measured the positions of approximatelly 500 000 stars. Several similar projects are currently in various stages of preparation, examples include the Pan-STARRS survey, Large Synoptic Survey Telescope (LSST), the JASMINE and Gaia space probes.

There are at least two major approaches to handle these large quantities of data. One is to recruit volunteers and have them analyse the observations over the Internet. Examples of such projects are the Galaxy Zoo project which used volunteers to classify resolved SDSS galaxies. The second approach is to use algorithmic data analysis, including various machine learning methods such as neural networks, support vector machines, clustering and principal component analysis.

Figure 1: A selection of various objects that an automated sky survey might encounter. Source: *"Classification and Discovery in Large Astronomical Surveys"* workshop poster.

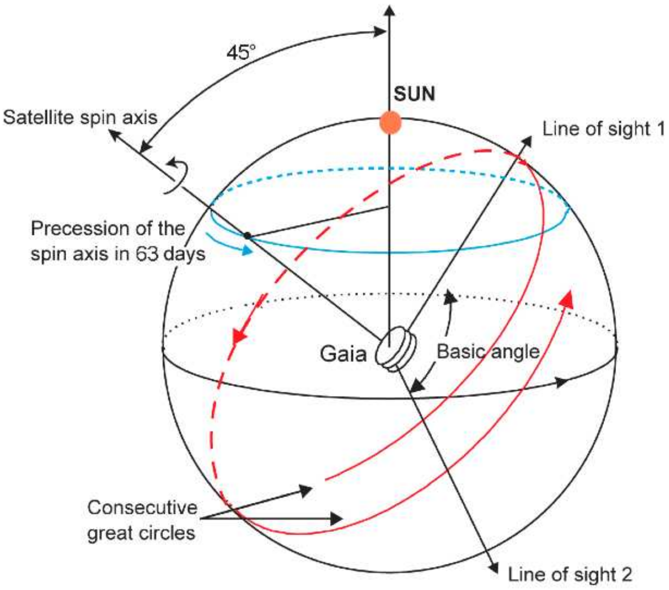

Figure 2: Scanning law of the Gaia space telescope. Source: [Ano06]

1.2 Gaia space telescope

The Gaia space telescope is an ambitious science endeavour of the European Space Agency (ESA) with the goal of measuring all objects in the celestial sphere up to the 20th magnitude, which corresponds to approximatelly 1 billion objects. [Ano06]

The satellite is currently under construction and is planned to launch in 2011 from a Soyuz-Fregat rocket from Guiana Space Centre. Following a succesful launch, Gaia will travel onto a stable orbit around the L_2 point of the Earth-Sun system[1]. There it will deploy it's telescope and scan the celestial sphere for the approximatelly five years.

The scanning law of the telescope is visualized in Figure 2. Gaia makes a full turn around it's axis in 6 hours. In addition, the spin axis of the satellite is slowly precessing, allowing Gaia to cover the whole sky in 63 days.

Gaia observes the sky using two telescopes (thereby improving positional measurements), which project the light collected onto a single CCD performing the measurements. A single object takes 4.4 seconds to pass over the CCD detector. A schema of the CCD is given in Figure 3.

- The Sky Mapper (SM) performs real-time object detection, determining which measurements of the latter components will be stored and transmitted to Earth.
- The Astrometric Field (AF) instrument then performs a precise positional measurement of the objects relative to each other.
- The Blue Photometer (BP) and the Red Photometer (RP) will measure low-resolution spectra of the object in the regions of 330–600 and 650–1000nm. This BP/RP Photometer data will be looked into in this thesis.

[1] One of five points, where the combined gravitational forces of the Sun and Earth allow a satellite to be in stationary orbit with respect to the Earth and the Sun. L2 lies directly "behind" Earth.

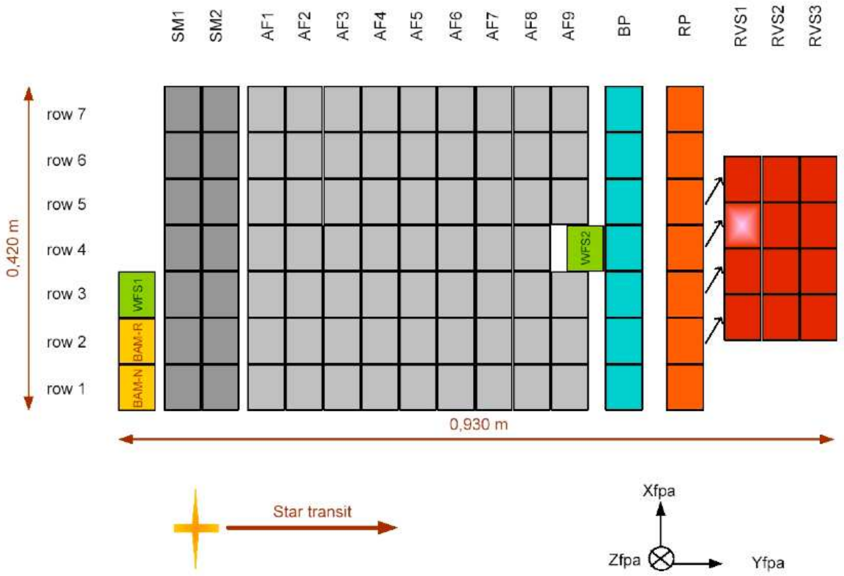

Figure 3: Schema of the Gaia CCD. Source: [Ano06], Figure courtesy of EADS Astrium.

- Finally, the Radial Velocity Spectrograph (RVS) will measure a relatively narrow area (847–874 nm) of the spectrum with a considerably higher resolution than the BP/RP Photometer. This measurement will only be performed for objects up to the 17th magnitude.

From the data analysis perspective, the amount of data produced by Gaia is formidable. although preliminary data reduction will be performed on-board and in real-time, the telescope will transmit approximatelly 100 Gigabits of compressed data to the ground station daily. After uncompressing and processing, the final database will have an estimated size of 1 petabyte [MR09].

The analysis of the dataset is also a formidable challenge in terms of sheer complexity. As an example, the accuracy of positional measurements is presise enough that the gravitational effects of the Sun bending starlight have to be taken into account in determining the real positions of the stars.

1.3 BP/RP Photometer

For our purposes, we can assume that light coming from a star consists of a non-polarized electromagnetic wave. A CCD element can be used to measure the amount of power that the wave carries, which lets us to obtain the *energy flux* F of the wave. This is measured in $W\,m^{-2}$, which is to be understood as "the power of a light measured on a sensor with a certain area".

Various techniques (e.g. glass prism, diffraction grid) can be used to spacially separate the different wavelengths, thus measuring the *flux density* (measured in $W\,m^{-2}\,nm$, which is to be understood as "energy flux of the light with a certain wavelength"). As stated earlier, measuring the energy distribution of the light (called a spectrum) coming from an astronomical body is usually the primary method of studying the object.

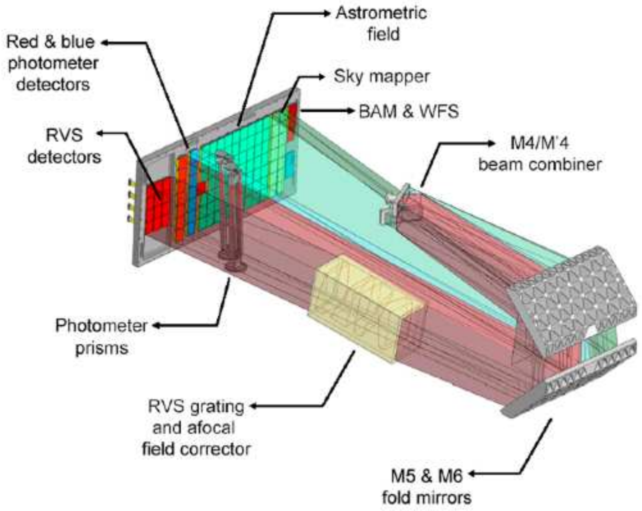

Figure 4: Photometric instrument of the Gaia space probe. Source: [Ano06], Figure courtesy of EADS Astrium.

The Gaia BP/RP Photometer, visualized in Figure 4, is a low-resolution prismatic photometer. Light entering the telescope will pass through a broadband filter that removes unwanted wavelenths. This is followed by passing through a fused-silica prism that disperses light according to wavelength. After this, the two light beams enter the CCD sensor and two spectra will be obtained, one in the blue region (330-660nm) and one in the red region (650-1000nm).

During it's 5 year mission, Gaia will measure a single object several times. Under certain conditions, these measurements may be combined to increase their accuracy. The exact details on how this will be done are not fixed yet, but it will probably be something similar to the drizzle algorithm that is currently being used in the Hubble Space Telescope [Smi08]. From now on we will refer to single-transit measurements as *epoch spectra* and to end-of-mission data as *combined spectra*.

1.3.1 Effective line width

A spectral line is a sudden peak (emission line) or sink (absorption line) in an otherwise smooth spectrum continuum, which can be approximated by a Gaussian function for our purposes. We use the measure of *effective line width* (aka *equivalent line width*) to charaterise spectral lines. This measure can be thought of as the equivalent wavelength interval over which the continuum area is equal to the area under the spectral line.

The effective line width is often used in astronomy for two reasons. First, it does not depend on the channel distribution (sampling) of the observation instrument. Second, it is insensitive to the distance of the star as well as to interstellar absorption (under certain assumptions). This allows us to compare spectral features of different objects without knowing their distance to Earth, which is often the case in astronomy due to technical limitations on distance measurements.

Formally, the effective line width is defined as follows. Let the whole spectra $F(\lambda)$

be represented as

$$F(\lambda) = F_l(\lambda) + F_c(\lambda)$$

where $F_l(\lambda)$ is the flux of a single spectral line and $F_c(\lambda)$ describes the flux of the continuum (everything else except the spectral line). The effective width of the spectral line is then [Sch06, page 181]

$$W = \int \frac{F(\lambda) - F_c(\lambda)}{F_c(\lambda)} \, d\lambda = \int \frac{F_l(\lambda)}{F_c(\lambda)} \, d\lambda$$

Assuming that the continuum under the spectral line is constant, one can think of this definition as the area of the spectral line divided by the height of the continuum at the center of the spectral line. Note that for an emission line, $W > 0$ and for an absorption line, $W < 0$. For calculating the effective line width integral from measured spectra, we use the *extended trapezoidal rule* [PTVF92, page 133]

$$\int\limits_{x_1}^{x_N} f(x) \, dx = h \left(\frac{1}{2} f_1 + f_2 + \ldots + f_{N-1} + \frac{1}{2} f_N \right) + O\left(\frac{(b-a)^3 f''}{N^2} \right)$$

1.3.2 Apparent magnitudes

The apparent magnitude is a practical convention for expressing the brightness of an object. Formally, the magnitude relation between two bodides is defined as [KKO+03, page 181]

$$m_1 - m_2 = -2.5 \log_{10} \left(\frac{F_1}{F_2} \right)$$

where m_1, m_2 are the respective magnitudes and F_1, F_2 are the physical fluxes discussed earlier. Note that the fluxes have to be taken over a fixed set of wavelengths and a specific object m_0 is usually defined to have zero magnitude to fix the scale.

To give examples, the apparent magnitude (in the visual band) of the Sun is -26.8 mag, the nearest star Sirius -1.5 mag and the nearest galaxy (Andromeda) 4.3 mag.

1.4 Emission line stars

We now give a short overview of emission line stars relevant to the data analysed in this thesis. It is important to note that this is not an exhaustive overview, but is intended to give some background on the data we are analysing.

1.4.1 Be stars

There are two types of Be stars, which we consider to be similar enough for the low-resolution Gaia BP/RP spectra to handle them as a single class. First, *classical Be stars* are rapidly rotating B type stars surrounded by a flattened disk of gas, which is the main cause of their H-α emission. Second, *Herbig Ae/Be* type stars are pre-main sequence stars often found near large molecular clouds. Somewhat more than 100 Herbig Ae/Be stars have been observed so far [SP04].

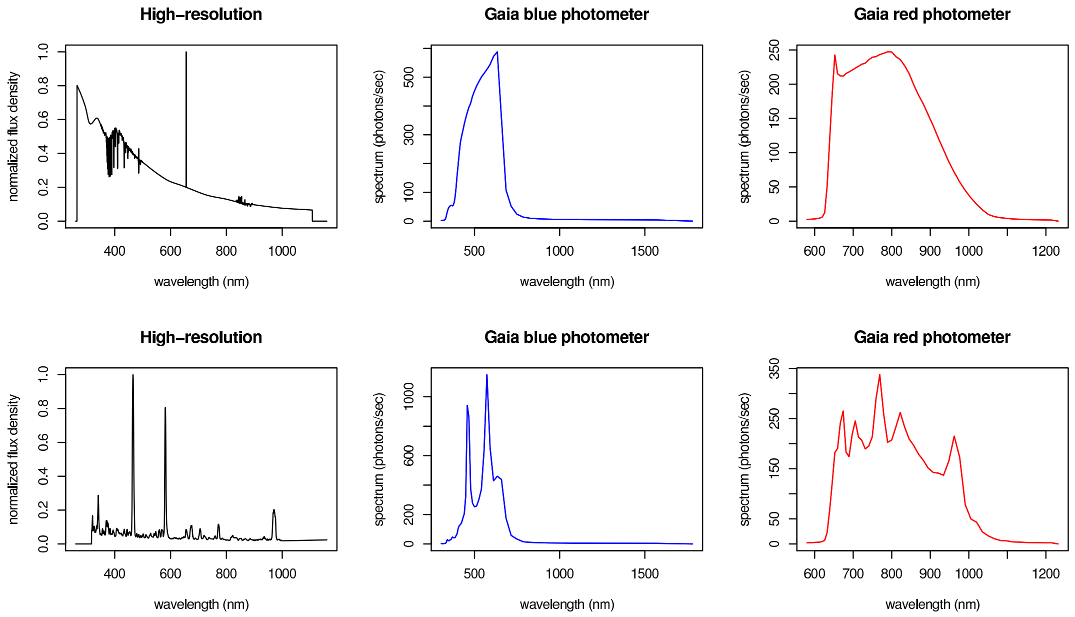

Figure 5: Examples of a Be star (top row) and Wolf-Rayet star (bottom row), both as high-resolution spectra and as Gaia BP/RP simulations.

An example of a Be star spectrum is given in the top row of Figure 5. In the high-resolution spectrum, on can observe the H-α emission line at approximately 650nm. One can also recognize an imprint of this emission line in the low-resolution BP and RP spectra.

1.4.2 Wolf-Rayet stars

Wolf-Rayet stars are a class of stars, whose spectra are dominated by broad emission lines, indicating an extreme mass loss [MoPGB01]. An example of a WR star spectrum is given in the bottom row of Figure 5, where two emission areas can be observed in the BP spectrum and six emission areas in the RP spectrum. The reasons behind the mass loss is an open problem in astrophysics, with various hypotheses attributing this to one or several factors from intense radiation pressure, a binary companion star or a peculiar stellar magnetic field.

Approximately 220 Wolf-Rayet stars have been observed in our Galaxy and a few hundred in neighbouring galaxies. Theoretical considerations on the total number of Wolf-Rayet stars in our Galaxy range between 1000 and 2000.

2 Computer Science Background

In this chapter, we give an overview of the computer science methods used in this thesis. First, we discuss Support Vector Machines, which are machine learning algorithms suitable for both classification (Section 2.1) and regression (Section 2.2). Our overview is mainly based on [CST04, Bur98].

In Section 2.3, we give a short summary of Principal Component Analaysis (PCA), which is a method for describing high-dimensional data using a linear combination of a small number of principal components. We later use PCA in Section 3.3 in constructing a model of stellar spectra for effective line width prediction.

2.1 Support Vector Classification

Introduction. Machine learning in general is concerned with finding a model describing a set of data, without any intrisic knowledge about the dataset. Formally, it is often defined in the following way. Given a set of data

$$S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}, \ \mathbf{x}_i \in X, \ y_i \in Y, \ i = 1, \ldots, l$$

find a function $f : X \to Y$ such that $f(\mathbf{x}_i) \approx y_i$ and f gives "reasonable" results on new unknown data not present in S. In case of a classification problem, Y is the set of class labels and for a regression problem, $Y \in \mathbb{R}$.

Maximal Margin Classifier. We begin the overview of Support Vector Machines by considering a two-dimensional classification problem. Formally this corresponds to $X = \mathbb{R}^2$ and $Y = \{+1, -1\}$. In addition, we assume that the data is linearly separable, meaning that there exists a line with the property that all points from one class are on one side of the plane and all points from the other class are on the other side of the plane. We can classify new unknown points by simply checking their position with respect to the line. Notice that there may be several lines that separate the two classes, so we would like to find a criteria that would uniquely define a line that separates the two points well. Additionally notice that the idea can be generalized to n-dimensional input data by using a hyperplane given by the equation.

$$\mathbf{w} \cdot \mathbf{x} + b = 0, \ \mathbf{w}, \mathbf{x} \in \mathbb{R}^n$$

The rule for determining the class y of a data point $\mathbf{x}$ is then given by the equation.

$$y = sign(\mathbf{w} \cdot \mathbf{x} + b)$$

We now continue to derive a way for uniquely defining a good hyperplane between two linearly separable point classes in n-dimensional space. Assume that $\mathbf{w} \cdot \mathbf{x} + b = 0$ is a separating hyperplane and let $\mathbf{x}_{+1}$ and $\mathbf{x}_{-1}$ be the closest points from the respective classes to the hyperplane. Choose a scale for $\mathbf{w}$ and b such that

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x}_{+1} + b &= 1 \\ \mathbf{w} \cdot \mathbf{x}_{-1} + b &= -1 \end{aligned}$$

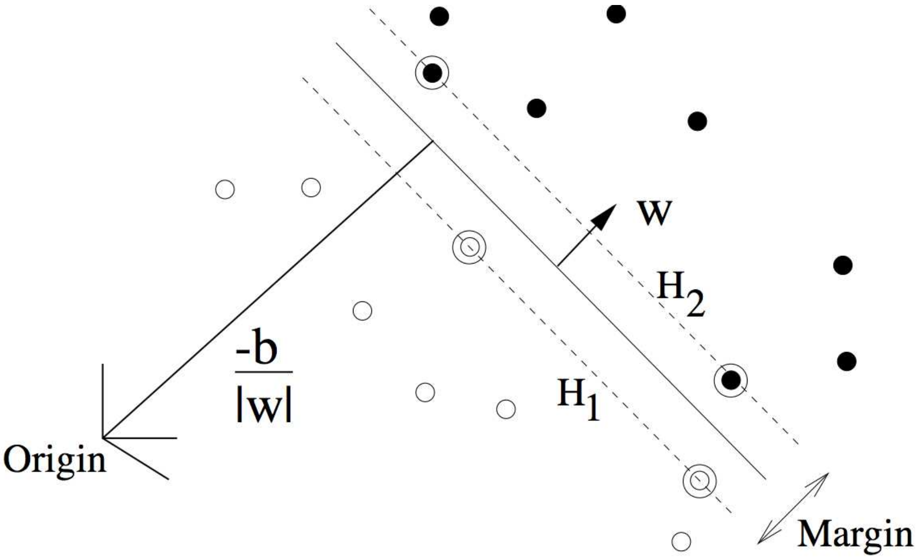

Figure 6: A hyperplane with the normal vector $\mathbf{w}$ separating two classes, represented by black and white dots. Source: [Bur98]

Recalling the fact from linear algebra that given a point $\mathbf{p} \in \mathbb{R}^n$ and a plane $\mathbf{w} \cdot \mathbf{x} + b = 0$, the directed distance d between this point and the plane is

$$d = d(\mathbf{p}, \mathbf{w}, b) = \frac{\mathbf{w} \cdot \mathbf{p} + b}{\|\mathbf{w}\|}$$

where $\|\mathbf{w}\| = \sqrt{\sum\limits_{i=1}^{n} \mathbf{w}_i^2}$, we now get for all data points

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x}_i + b &\geq 1, \ y_i = +1 \\ \mathbf{w} \cdot \mathbf{x}_i + b &\leq -1, \ y_i = -1 \end{aligned}$$

which we can combine into

$$y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) \geq 1, \ i = 1, \ldots, l$$

Let $d_{+1} = d(\mathbf{x}_{+1}, \mathbf{w}, b)$ and $d_{-1} = d(\mathbf{x}_{-1}, \mathbf{w}, b)$ be the distances between the hyperplane and the points $\mathbf{x}_{+1}$ and $\mathbf{x}_{-1}$. We now define the margin M of the hyperplane as $M = d_{+1} + d_{-1}$. In order to find the hyperplane that best separates the classes, we now decide to select the hyperplane with the maximal margin M between the classes. This is illustrated on Figure 6.

Since the margin M is the projection of the distance between $\mathbf{x}_{+1}$ and $\mathbf{x}_{-1}$ onto any vector perpendicular to the hyperplane (and $\mathbf{w}$ is obviously perpendicular to the hyperplane), we can calculate M as follows.

$$\begin{aligned} M &= pr_{\mathbf{w}} \left(\mathbf{x}_{+1} - \mathbf{x}_{-1} \right) = \frac{\mathbf{w} \cdot \left(\mathbf{x}_{+1} - \mathbf{x}_{-1} \right)}{\|\mathbf{w}\|} = \frac{\mathbf{w} \cdot \mathbf{x}_{+1} - \mathbf{w} \cdot \mathbf{x}_{-1}}{\|\mathbf{w}\|} = \\ &= \frac{1 - b - (-1 - b)}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|} \end{aligned}$$

Thus, maximizing the margin M is equivalent to minimizing $\|\mathbf{w}\|$ while requiring all the points from both of the classes to lie on the correct side of the hyperplane and outside the margin, represented by the planes H_1 and H_2 in Figure 6. Formally, we have the following optimisation problem.

$$\begin{aligned} &\min_{\mathbf{w},b} \mathbf{w}\cdot\mathbf{w}, \\ \text{subject to}\quad & y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) \geq 1, \quad i = 1,\ldots,l \end{aligned}$$

This classifier obtained from this optimisation problem is called the *Maximal Margin Classifier.*

Soft Margin Classifier. Data in the real world is often noisy. Thus, it may be possible that some of the points in the set S are wrong, which may have adverse effects to the ability of the classifier to function properly. To improve the situation, we allow some points to lie on the incorrect side of the margin by introducing the so-called slack variables ξ_i into the optimisation problem. Formally, one way to introduce these slack variables is

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \mathbf{w}\cdot\mathbf{w} + C\sum_{i=1}^{l}\xi_i, \\ \text{subject to}\quad & y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0,\ i = 1,\ldots,l \end{aligned}$$

Here, the variable C is a free parameter representing the tradeoff between model complexity and accuracy, often called the *complexity constant.* In practice, it is usually determined experimentally by repeatedly learning the model with different values of C and then selecting the value which results in the best performance. The classifier obtained from this optimisation problem is called the *Soft Margin Classifier.*

Finding the optimal hyperplane. Using results from optimisation theory, it is possible to show that the Soft Margin Classifier optimisation problem is equivalent to the following optimisation problem.

$$\begin{aligned} \max_{\alpha}\quad & W(\alpha) = \sum_{i=1}^{l}\alpha_i - \tfrac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \mathbf{x}_i\cdot\mathbf{x}_j, \\ \text{subject to}\quad & \sum_{i=1}^{l} y_i\alpha_i = 0, \\ & C \geq \alpha_i \geq 0, \qquad i = 1,\ldots,l \end{aligned}$$

with the decision rule on determining the class of the point $\mathbf{x}$ taking the form

$$y = sign\left(\sum_{i=1}^{l} y_i\alpha_i^{*}\mathbf{x}_i\cdot\mathbf{x} + b^{*}\right)$$

It is possible to solve this optimisation problem using numerical methods that normally converge to the global optimum. There are several libraries available for specifically solving these kinds of optimisation problems. We use the LIBSVM implementation in our experiments [CL01].

Classifying non-linear data with kernels. Until now, we have assumed that the data points in S are linearly separable. In order to classify non-linear data, we use a mapping $\phi(\mathbf{x})$ to project the data into an another space called *feature space* with the idea that it may be linearly separable in the feature space, i.e.

$$\mathbf{x} = (x_1, \ldots, x_n) \longmapsto \phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_N(\mathbf{x}))$$

Usually, it is good to have a feature space with a large dimensionality, to increase the chances of the data becoming linearly separable.

By noticing that the Soft Margin Classifier optimisation problem only computes with scalar products of the input data, we can define the following function $K(\mathbf{x}, \mathbf{z})$, which is called a *kernel function*

$$K(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z})$$

By replacing the scalar products of the input data with the kernel function in the Soft Margin Classifier, our optimisation problem takes the form

$$\begin{array}{lll} \max\limits_{\alpha} \quad W(\alpha) = \sum\limits_{i=1}^{l} \alpha_i - \frac{1}{2} \sum\limits_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j), & & \\ \text{subject to} & \sum\limits_{i=1}^{l} y_i \alpha_i = 0, & \\ & C \geq \alpha_i \geq 0, & i = 1, \ldots, l \end{array}$$

with the decision rule on determining the class of the point $\mathbf{x}$ changing to

$$y = sign\left(\sum_{i=1}^{l} y_i \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*\right)$$

It is this variation of the Soft Margin Classifier with a kernel function that is usually understood as the *Support Vector Machine* (SVM) algorithm.

There are several different kernel functions to choose. One of the most widely used is the Radial Basis Function (RBF) kernel, which LIBSVM defines as

$$K_{RBF}(\mathbf{x}, \mathbf{z}, \gamma) \quad = \quad \exp\left(-\gamma \left\|\mathbf{x} - \mathbf{z}\right\|^2\right), \; \gamma > 0$$

The RBF kernel has a free parameter γ, which is usually determined experimentally like the complexity constant. We use the RBF kernel in our experiments.

Generally, the use of kernels can further widen the opportunities of applying SVMs by defining kernels on non-numeric data (e.g. protein primary structure [MIK$^+$07]).

Classifying more than two classes. Until now we have only considered a binary classification problem. The most often used practical method of using SVMs on a classification problem with more than two classes is to train several binary classifiers on different classes and then combine the results. There are several different ways of doing this. LIBSVM uses the "one-against-one" approach, which works by training $k(k-1)/2$ binary classifiers for a classification problem with k classes. Each binary classifier is trained only using data from two classes from the data. In determing the class, each classifiers votes for one (out of two possible) classes. The class which received most votes is assigned to the data point.

2.2 Support Vector Regression

The ideas behind the SVM algorithm can also be applied to regression problems where instead of discrete classes, $y \in \mathbf{R}$. One possible way of formulating the regression problem is the ϵ-Support Vector Regression (SVR), which is formulated as follows [CST04, page 117]

$$\begin{aligned} & \min_{\mathbf{w},b,\xi_i,\hat{\xi}_i} \|\mathbf{w}\|^2 + C\sum_{i=1}^{l} \left(\xi_i + \hat{\xi}_i\right) \\ \text{subject to} \quad & (\mathbf{w} \cdot \phi(\mathbf{x_i}) + b) - y_i \leq \varepsilon + \xi_i, \quad \xi_i \geq 0,\ i = 1, \ldots, l \\ & y_i - (\mathbf{w} \cdot \phi(\mathbf{x_i}) + b) \leq \varepsilon + \hat{\xi}_i, \quad \hat{\xi}_i \geq 0,\ i = 1, \ldots, l \end{aligned}$$

The optimisation problem is formulated such that differences between the data and the model that are smaller than ε do not affect the model. Here, the constant C again determines the tradeoff between model complexity (a high C value results in a larger penalty for allowing the difference between a data point and the model to exceed ε).

Support Vector Regression can be used for learning to predict complex non-linear variables only based on examples. For example, preliminary results have shown good performance in predicting various astrophysical parameters from (simulated) unresolved galaxy spectra of the Gaia space probe [TKBJ+07].

2.3 Principal Component Analysis

Principal Component Analysis (PCA), also known as the *Karhunen-Loéve transform*, is an algorithm for representing high-dimensional data in terms of a small number of principal components that describe the data as well as possible [Bis06]. In particular, given a set of m-dimensional observations $\mathbf{x}_i$, $i = 1, \ldots, n$, PCA finds m-dimensional principal components such that the projection of the input data on the i-th component has the i-th largest variance (meaning that the first principal component has the largest variance). Computationally, one way of determining the principal components involves calculating the eigenvectors and eigenvalues of the covariance matrix of the dataset.

PCA has several uses, including data compression and dimensionality reduction. We use the standard `prcomp()` routine from the R `stats` package in our experiments.

3 Materials and Methods

In this chapter, we describe the experiments performed on detecting emission line stars from Gaia spectra. We give an overview of the data that we used, including the synthetic spectral model constructed for the effective line width experiments. Also, we briefly describe the Gaia Object Generator used for obtaining the simulations of the BP/RP Photometer observations.

3.1 Datasets

Gaia Object Generator. To facilitate the construction of the data reduction pipeline for Gaia, several simulators have been created to construct spectra resembling real Gaia observations from existing high-resolution observed or modelled spectra. The *Gaia Object Generator* (GOG) [ZBI$^+$08, IZS$^+$08] is one of such simulators. In our work, we used GOG v5.01, which produces so-called cycle 3 data. When generating combined spectra using GOG, we used the default number of 80 transits and a sampling factor of 3.

Local database. We used an existing dataset of 72 high-resolution spectra from the Tartu Observatory, consisting of various types of Be stars, Wolf-Rayet (WR) stars and nebulae. In many cases, the whole spectrum from 260nm to 1160nm was not represented, we approximated with constant interpolation where necessary.

CU8 cycle 3 simulated libraries. The Astrophysical parameters group CU8 of DPAC is in the process of developing modelled spectral libraries of various astronomical objects, including main-sequence stars, Be stars and WR stars [DPA]. We use this data in training our classifier in the classification task.

3.2 Classification

In the direct classification task, we are trying to find an algorithm that separates spectra into discrete classes. For the scope of this thesis, we are going to make two simplifications. First, we are going to assume that the previous data reduction pipeline has already removed all other spectra so that we are only dealing with stellar spectra as input. Second, we are assuming that we only have two kinds of emission line stars, namely Be and WR stars in our data. Our previous work [JLK08] has demonstrated the feasibility of using Support Vector Machines for separating Be-stars with a H-α emission from regular main sequence stars.

Available datasets are visualized on the following page. For Be stars we have 174 CU8 models and 15 local observations, see Figure 7. In the case of WR stars, we have 9 CU8 models and 25 local observations, see Figure 8. Finally, we have an abundance of 20180 CU8 simulations for regular stars, see Figure 9. To limit the effects of class skew, we are going to undersample regular stars in our experiments.

There are notable differences between the CU8 models and local observations. For Be stars, CU8 models show an emission line at approximatelly 450nm which is not recognizable in the observations. In case of WR stars, same emission lines are mostly

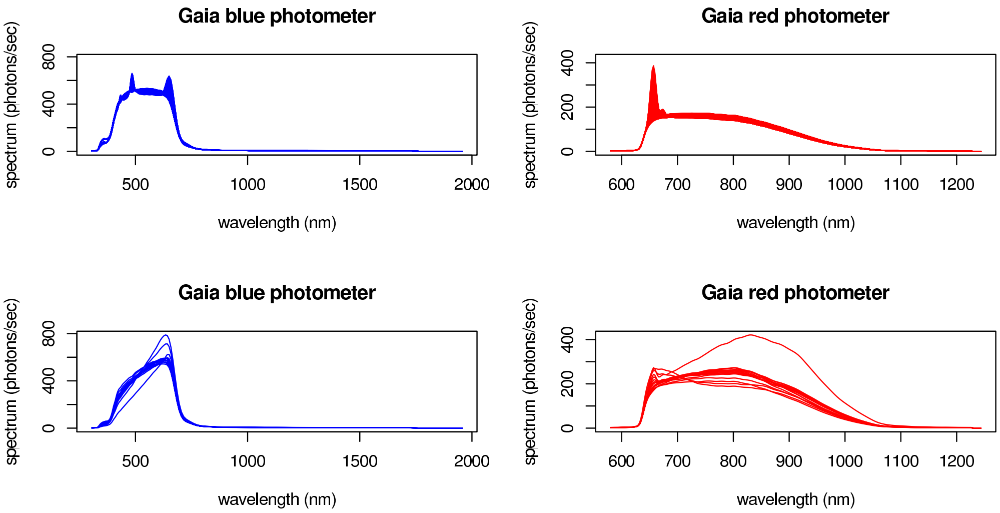

Figure 7: Gaia simulations of Be stars from CU8 models (top) and local observations (bottom).

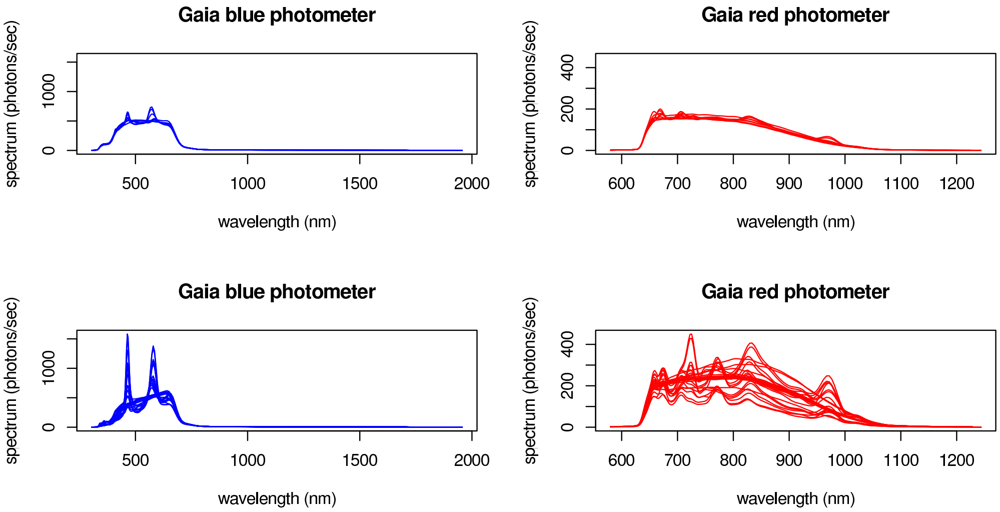

Figure 8: Gaia simulations of WR stars from CU8 models (top) and local observations (bottom).

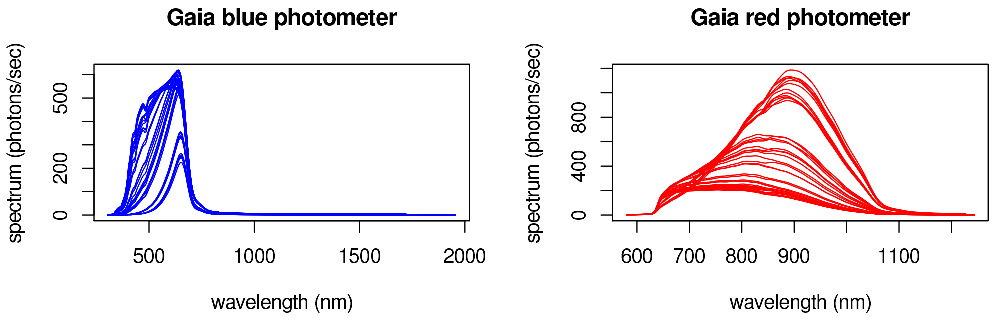

Figure 9: Gaia simulations of regular stars from CU8.

present in both CU8 models and local observations, but the models have clearly weaker emission than observations.

Ideally, we would like to train our classifiers on modelled data, that covers the whole possible class space in a uniform manner. This approach has achieved good results in e.g. classifying galaxies [TKBJ+07]. although available emission line star models are quite limited, both in terms of quality and quantity, we are going to try to train our classifiers on modelled data and validate on observed data.

We additionally perform a second detection experiment and cross-validate our classifiers on observed data. In this case, training data will be more similar to testing data. Thus, the cross-validation experiments should provide a rough estimate on the performance of a classifier when trained on hypothetical modelled spectra that more closely match actual observations. Due to the low number of observations available, this is likely to be a pessimistic estimate.

We approach the classification as two binary classification problems by first separating stars with H-α emission and then dividing these into Be stars and WR stars. This will allow us to use different features for the two problems, i.e. we can only use wavelengths around the 656nm in detecting H-α emission. Additionally, binary classification will allow us to discuss the results without going into the issue of the rarity of emission line stars in actual Gaia observations, which lies beyond the scope of the current work.

3.2.1 Detecting H-α emission

At this point, we are only interested in detecting emission at a specific wavelength, so we are going to select only channels with wavelengths in the range from 600nm to 700nm as input. This results in 15 features for epoch spectra and 45 features for combined spectra. As discussed earlier, we are going to study detection with two different kinds of experimental set-ups. First, we *train our classifier on modelled data* and *validate on observed data* (while still being forced to use CU8 models for regular stars in validation). In particular, we proceed as follows.

Training data. For objects with H-α emission, we use a combination of 174 CU8 Be models and 9 CU8 WR models. For objects with no H-α emission, we use a randomly drawn sample of 183 CU8 models. This undersampling allows us to avoid the possible problems of classification bias that can occur when training an SVM with unequal class proportions in the training data.

Validation data. For objects with H-α emission, we use a combination of 15 local observations of Be stars and 25 local observations of WR stars. For objects with no H-α emission, we use a randomly drawn sample of 40 CU8 models that has no overlap with the selection of regular star spectra in the training data.

Parameter tuning. We perform parameter tuning by conducting a relativelly coarse but exhaustive full-grid search on the complexity constant C and the constant γ of the RBF kernel ($C = 2^{-3}, 2^{-2} \ldots, 2^{3}$ and $\gamma = 2^{-8}, 2^{-7}, \ldots, 2^{-2}$). The meaning of these constants were described in Section 2.1.

Training and validation. We then train the classifier with the optimal parameters found using all training data. Finally, we validate the optimal classifier on the validation data.

Second, we perform *cross-validation with local observations* (and, again, CU8 simulations for regular stars) to assess the performance of the classifier. We do this to roughly estimate how the discrepancies between the CU8 models and observed spectra affect the classification accuracy. Formally, we take the following steps.

Cross-validation data. For objects with H-α emission, we use a combination of 15 local observations of Be stars and 25 local observations of WR stars. For objects with no H-α emission, we use a randomly drawn sample of 150 CU8 models.

Cross-validation. We divide the data D into n parts D_i. We then train n classifiers, each time using $D - D_i$ as training data and D_i as validation. As in the previous case, we do this over a set of free parameters. After some experimentation, we picked $n = 3$ as the number of folds due to needing as many points as possible from a single cross-validation run to plot averaged ROC curves for presenting our results.

Aggregation. Finally, we present the aggregated predictions accross the n experiments on the best combination of free parameters as results.

3.2.2 Separating Be stars and Wolf-Rayet stars

In separating Be stars and WR stars, we use all channels in the RP spectrum as input, which corresponds to 60 features for epoch spectra and 180 features for combined spectra. We perform similar experiments as in the case of H-α emission detection. We use identical setups to the H-α classification by both training on CU8 models with validation on local observed data and cross-validation on local observed data. For the first experiment, we define our classes as follows.

Training data. For Be stars, we use 174 CU8 Be models. For WR stars, we use 9 CU8 WR models.

Validation data. For Be stars, we use 15 local observations and for WR stars, we use 25 local observations.

For cross-validation with local observations, we use the data as follows.

Cross-validation data. For Be stars, we use 15 local observations and for WR stars, we use 25 local observations.

The results of our classification experiments are presented in Section 4.1.

3.3 Synthetic spectra

In the classification task, we performed the experiments on existing data. In the case of estimating the effective width of a single spectra line, we are going to use a naive model introduced in this chapter to study line characterization. When trying to characterize individual spectral lines, we cannot use the CU8 models because we do not have the corresponding high-resolution data to train the detection algorithms. Thus we are mostly going to rely on a naive model introduced in this chapter to

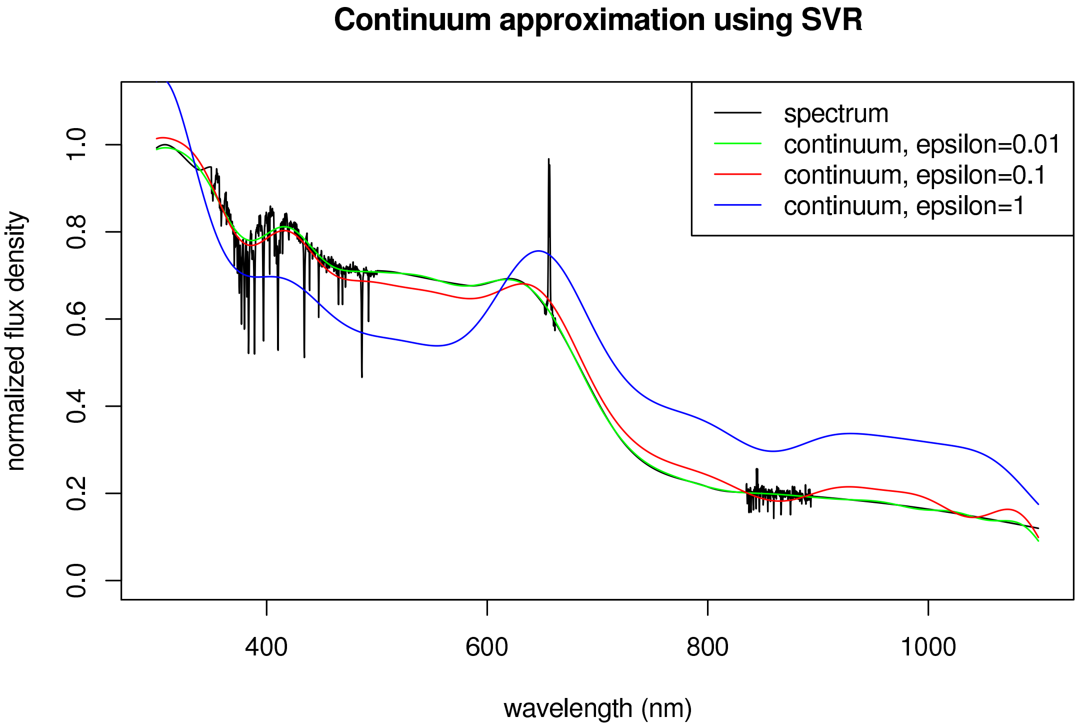

Figure 10: Approximating the continuum from a high-resolution spectrum using SVR with different ε-parameters.

study line characterisation. It is important to note that the naive model should be thought of as simply an algorithm for generating data similar enough to real spectra *only for the purpose of characterising individual spectral lines.* It does not conform to an astrophysical model of the star that simulates the underlying physical proccesses that form the spectrum.

Overview of the synthetic model. For our purposes, a spectrum of a star can be described by sum of a smooth continuum $C(\lambda)$ which doesn't have a good analytical approximation and a number of spectral lines $L_i(\lambda)$ which can be approximated by a Gaussian curve. Formally

$$F(\lambda) = C(\lambda) + \sum_i L_i(\lambda)$$

where $F(\lambda)$ is the spectrum, $C(\lambda)$ is the continuum and $L_i(\lambda)$ represent the spectral lines. These are further defined as

$$L_i(\lambda) = A \exp\left(-\frac{(\lambda - \lambda_0)^2}{2\sigma^2}\right)$$

where A determines the height, λ_0 the central wavelength and σ the width of the spectral line. Note that the spectral line can be an absorption line ($A < 0$) or an emission line ($A > 0$).

Estimating the continuum. We would like to estimate the continuum spectrum $C(\lambda)$ on the basis of a set of existing high-resolution stellar spectra. For this, we first need to extract the continuum from the stellar spectra, subtracting spectral lines in the process. To achieve this goal, we use support vector regression with a suitable ε-tube size. Recall from Section 2.2 that the SVR algorithm constructs a model that ignores errors that are smaller than ε. In addition the algorithm is allowed to tolerate single "outliners" (such as spectral lines) at the cost of a simpler model. Thus, selecting a sufficiently small ε allows us to approximate the smooth continuum. The behaviour of the SVR algorithm on various ε values is given in Figure 10.

To obtain the continuum, we first train the SVR with the wavelengths as inputs and the respective spectral fluxes as outputs. We then evaluate the SVR model on the required wavelengths, thereby obtaining the continuum.

Generating the continuum. Since the detection algorithms should be insensitive to the general shape of the continuum, we would like to generate as much variation in the continuums of our synthetic models as possible. We also do not have an analytical expression of the continuum, meaning that we can only use a set of existing numerical examples as input. Our solution is to generate a synthetic continuum by taking a random linear combination of the PCA components of the set of continua.

Formally, assume we have n continuum spectra C_i, each having m discrete channels. Running a PCA analysis for this set will give us n principal components P_i (m-dimensional) and a matrix of weights w_{ij} such that

$$C_i(\lambda) = \sum_{j=1}^{n} w_{ij} P_j(\lambda)$$

where w_{ij} are arbitrary weights and principal components with lower indices tend to describe more variation in the original dataset.

To generate random continuum C_{rnd}, we take a linear combination of the PCA components

$$C_{rnd}(\lambda) \quad = \quad \sum_{j=1}^{n} N(\mu_j, \sigma_j^2)\, P_j(\lambda)$$

where $N(\mu_j, \sigma_j^2)$ is a random function with a Gaussian distribution. The parameters μ_j and σ_j^2 are approximated from the original weights $w_{1j} \dots w_{nj}$.

It is important to note that we need to keep the train and validation sets different. Thus we divide our continuum data into two groups, one will be used to generate the PCA components for the training data and the other one will be used for the testing data. Additionally, there are cases where the generated random spectrum has some negative fluxes. We simply discard such spectra.

Generating spectral lines. The spectral lines are simply approximated using the standard Gaussian function $L_i(\lambda)$. The free parameters A and σ are approximated using a uniform distribution. The wavelength λ_0 is selected from physical considerations—we take $\lambda_0 = 656.28nm$ for the H-α. As in the case of continuum generation, we simply discard a spectrum if a spectral line causes negative fluxes.

To summarize, we use the following steps to obtain a synthetic spectrum.

1. Extract continuum component from a sample of existing spectra.

2. Run PCA on the set of continuum spectra.

3. Generate a random continuum $C(\lambda)$ from a random linear combination of the PCA components.
4. Add the necessary spectral lines $L_i(\lambda)$ to the spectrum.

3.4 Effective width of a fixed spectral line

Our goal is to estimate the effective width (see Section 1.3.1) of a spectral line at a *known, fixed wavelength.* We are going to limit ourselves to studying the H-α line at 656nm, although the results should be transferrable to other spectral lines. For example, in the case of WR stars emission lines at 470nm and 580nm for BP and 715nm and 770nm for RP are of interest.

Our previous results [JLK08] show that the Support Vector Regression algorithm is a stable algorithm useful for characterising emission lines from combined spectra. Using epoch spectra reduced the performance noticeably.

In a major improvement over our previous results we are now using the synthetic spectral model described in Section 3.3 for studying our algorithm. In addition to the sheer increase in training data, we also use principal components from different types of continua (Be stars, Wolf-Rayet stars and nebulae), thus making the algorithm applicable to a larger selection of objects. In a further attempt to make validation more realistic, we generate our training data and validation data from different, independant subsets of our spectra. In particular, we use the following steps in our experiments.

Generating the data. We start by generating 800 high-resolution training spectra T and 200 high-resolution validation spectra V using the model described in Section 3.3. Then we calculate the true effective line width from the high-resolution spectra as described in Section 1.3.1. Finally, we simulate low-resolution Gaia measurements from the high-resolution spectra as outlined in Section 3.1.

Parameter tuning. We divide the training data T into two groups T_1 of size 600 spectra) and T_2 of size 200 spectra. We perform parameter tuning by conducting a relatively coarse but exhaustive full-grid search on the complexity constant C and the constant γ of the RBF kernel $C = 2^{-5}, 2^{-4} \ldots, 2^{15}$ and $\gamma = 2^{-15}, 2^{-14}, \ldots, 2^{3}$. These are described in Section 2.2.

Training and validation. We then the train the algorithm using all training data T with the best combination of free parameters found. Finally, we validate this model on validation data V.

The results of our line width estimation experiments are presented in Section 4.2.

4 Results and Discussion

In this chapter, we present the results of our detection experiments. In Section 4.1 we discuss the classification problem, followed by the line width problem in Section 4.2. In Section 4.3 we try to compare these two approaches and draw a few conclusions on the direction of future work.

4.1 Classification

In this section we start with detecting H-α emission and continue with separating Be and WR spectra. All spectra are simulated as magnitude 15. Gaia is planned to observe all objects up to magnitude 20, but the CU8 cycle 3 simulations were limited to magnitude 15. We expect objects with greater magnitude to have a degraded signal to noise ratio and reduced performance. Thus, our results can be interpeted as the lower limit for objects of magnitude 15 and lower.

We use *Receiver Operating Characteristics* (ROC) graphs to visualise our classification results[Faw04]. ROC graphs can be used with classifiers that output a continuous decision value indicating the hypothesized class. Discrete classes are then assigned from a fixed treshold value for this decision score. For the SVM, the decision value is usually taken to be the distance of the data point from the optimal hyperplane. A ROC curve is a plot of the false positive rate to the true positive rate in dependence of the threshold decision value. An ideal classifier would be in the upper left corner of the plot and a random classifier would lie on the diagonal line. ROC curves have the useful property of being insensitive to the underlying class distribution. This is desirable, since we want to ignore the problems of the rarity of emission line stars for the moment. It can be shown that the total area under the ROC Curve (AUC) is equal to the likelihood of the classifier ranking a randomly chosen positive instance higher than a randomly chosen negative instance. We use the `ROCR` R package for generating the ROC curves [SSBL05].

4.1.1 Detecting H-α emission

We selected channels in the range of 500nm...600nm (around the H-α line) as input. The performance of the SVM algorithm in predicting H-α emission is given in Figure 11. On the left, we trained using CU8 model spectra and validated with local observations. On the right, we performed cross-validation on local measured spectra.

Training on modelled spectra results in an excellent classifier when using epoch spectra. The huge drop in classifier performance when switching to combined spectra is unexpected. We have checked our data handling routines for bugs, without success. Thus, the explanations for this strange behavior will be left to be explained in future work.

Cross-validation on observed spectra gives excellent results both in the case of epoch spectra and combined spectra. Considering the small discrepancies in the AUC measures, the differences in the models do not seem to have a significant impact on detection accuracy.

Since the differences between the performances for epoch spectra and combined

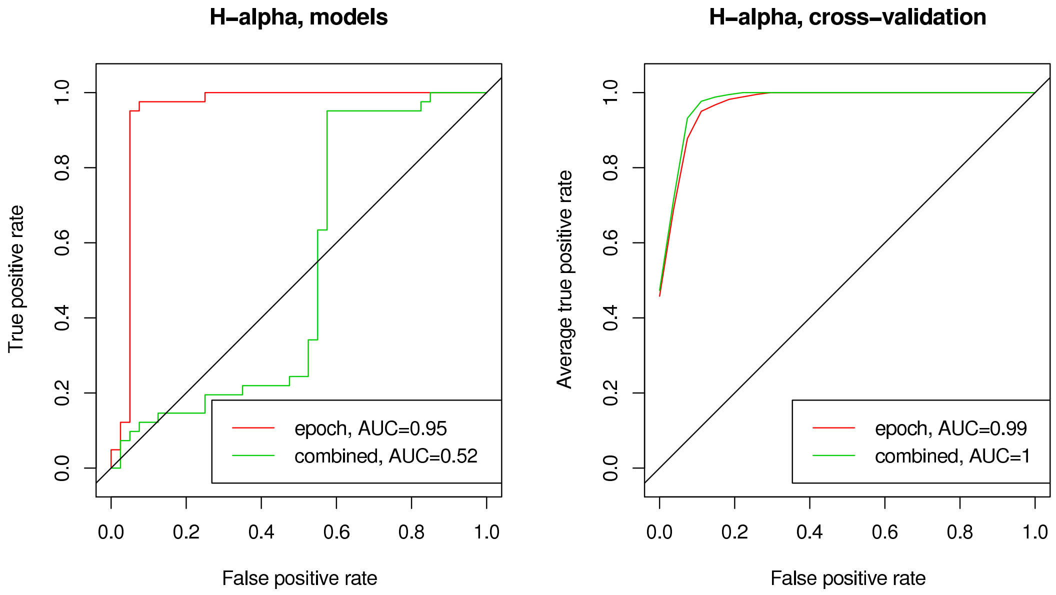

Figure 11: Performance of the SVM in detecting H-α emission. Left: training on modelled spectra, validation on measured spectra. Right: cross-validation on measured spectra.

spectra are neglible in the case of cross-validation, we hypothesize that using combined spectra does not have a significant impact on H-α detection accuracy, at least for objects up to magnitude 15. This is an important result, given the fact that combined spectra can only be obtained after the 5-year mission of Gaia is over, whereas epoch spectra can be obtained significantly sooner.

4.1.2 Separating Be stars and Wolf-Rayet stars

The performance of the SVM algorithm in separating Be stars and Wolf-Rayet stars is given in Figure 12. As in the H-α prediction experiment, the left graph depicts training using CU8 model spectra and validation with local observations. The right graph depicts cross-validation on local measured spectra.

We are dealing with a small number of data points, meaning that the results have a large uncertainty attached. The overall performance is quite poor, which may be explained by the fact that there simply isn't enough data for the SVM algorithm to construct a proper model. For the most extreme case, recall from Section 3.2 that on training with modelled data, we only have 9 samples of WR stars in the training data.

Acknowledging this, we would like to present two hypotheses. First, using combined spectra does seem to improve separation accuracy. Second, cross-validation does have better performance in terms of the AUC scores, both in the cases of epoch data and combined data. We emphasize that validity of these statements will have to be checked in future work that hopefully includes more data.

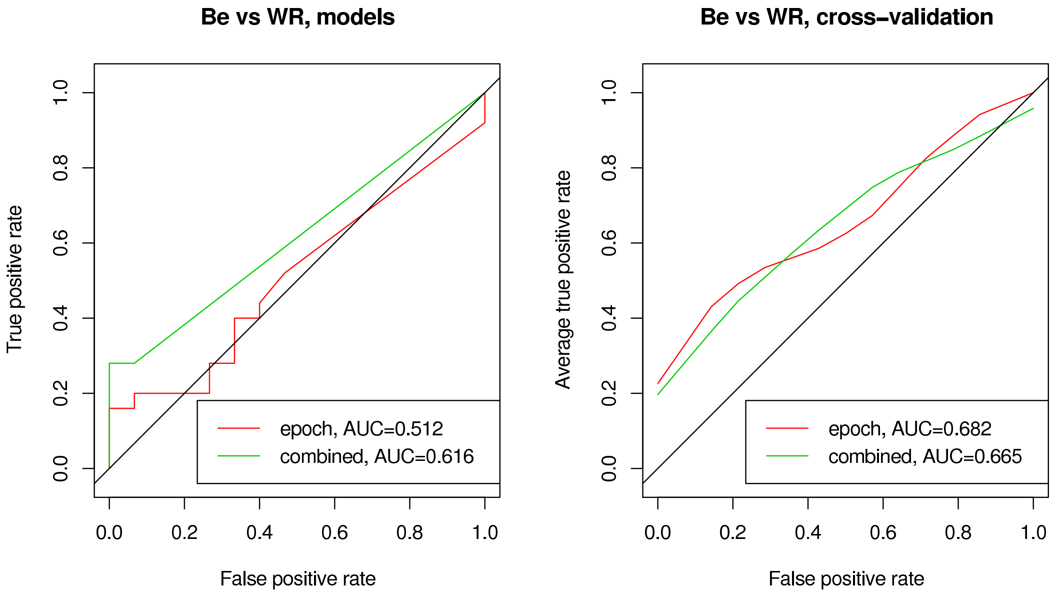

Figure 12: Performance of the SVM in separating Be and WR spectra. Left: training on modelled spectra, validation on measured spectra. Right: cross-validation on measured spectra.

4.2 Line-width estimation

We now present the results on estimating the effective line width of the H-α line from the BP/RP spectra. We look at how apparent magnitude and oversampling effect the predicition accuracy.

We used a training set of 800 spectra as well as a validation set of 200 spectra, obtained using the model described in Section 3.3. Both datasets were simulated at three different magnitudes using the GOG as described in Section 3.1. For the parameter tuning phase, we randomly separate the training set into two parts, using one of them for training and the other for testing as described in Section 3.4.

We present our results as scatter plots of true line width on the horizontal axis and predicted line width on the vertical axis. In addition, we also visualize the classifier performance as *Regression Error Characteristic* (REC) curves [BB03]. REC curves are plots of an error measure on the horizontal axis and the fraction of predictions within the given error on the vertical axis. One can plot different error measures on the horizontal axis, we use the absolute error. Thus, the ideal classifier would have an REC curve of a straight line through the point $(0, 1)$ parallel to the horizontal axis.

We first experimented with using clean, noiseless spectra as training data, hoping that using clean training data would allow the SVR algorithm to construct a more accurate model, thus improving prediction performance even on noisy spectra. The results of this experiment, using combined spectra at magnitude 15, are presented in Figure 13. Using clean spectra as training data results in degraded performance which may be explained by *overfitting*, meaning that the algorithm is not able to generalize the input data to the noisy test data. We use noisy training spectra in all further experiments.

Since we used our own synthetic model to generate the data sets, we were able to generate Gaia observations on different magnitudes. We now look at the effects that

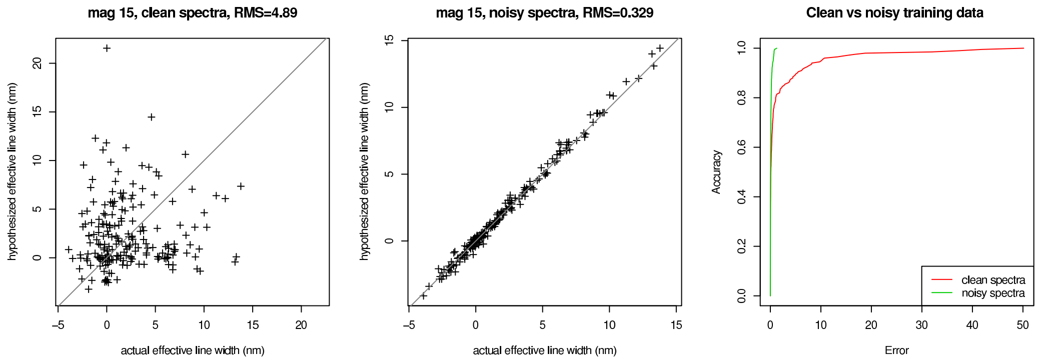

Figure 13: Effect of noiseless training spectra.

magnitude has on effective line width estimation. The results of our experiments are represented as REC-curves on Figure 14 and as comparison plots on Figure 15. Our results agree with the quite reasonable hypothesis that objects with a higher magnitude produce a substantially smaller signal-to-noise ratio, thus making the estimation of the effective line width more difficult. Still, even at magnitude 20 (which is the detection limit of the BP/RP instrument), the SVR algorithm is capable of satisfiably inferring the effective line width from combined spectra.

We again look at Figures 14 and 15 to study the difference between using combined spectra and epoch spectra. Combined spectra show a clear advantage over epoch spectra in predicting the H-α effective line width, with results from magnitude 20 combined spectra being comparable to epoch spectra from magnitude 17.5. This is consistent with our earlier experiments from [JLK08].

To conclude, our experiments demonstrate the feasibility of using the SVR algorithm to infer the effective line width of a spectral line from the low-resolution BP/RP spectra. Future work should proceed with validation on real spectra.

4.3 Discussion

In brief, the results of the H-α experiment were encouraging. Even with limited data, the SVM algorithm was able to achieve high accuracy in H-α emission detection. The next step in H-α detection experiments should be to repeat them at greater magnitudes to see wether the increased signal-to-noise ratio affects the accuracy of the classifier. Additionally, future experiments should take into account the extreme rarity of emission line stars compared to regular stars as discussed in Section 1.4.

The results for separating Be stars from WR stars were less encouraging, hinting at the need for larger datasets both for training and testing. One way to partially remedy this problem and perhaps improve the results would be to use an oversampling algorithm, such as SMOTE [CBHK02].

The results on using SVR in effective line width estimation are promising. One of the main reasons for this is the ability to use a sythetic model for generating our dataset. Given enough training data, SVR seems to be able to infer the effective line width quite accuratelly. Thus, using SVR as a classifier for i.e. H-α emission would consist of the trivial step of defining a reasonable cutoff for the line width. Additionally, by adding continuum spectra from other astronomical bodies to the synthetic model,

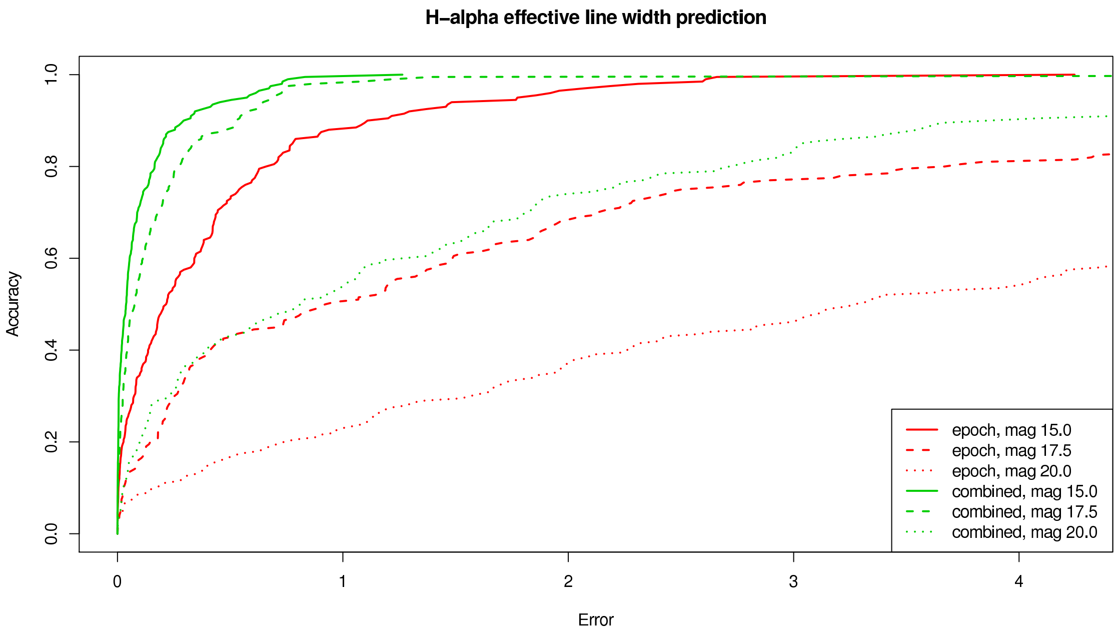

Figure 14: Performance of H-α effective line width prediction on three different magnitudes for epoch spectra (red) and combined spectra (green). A larger area under the Error-Accuracy curve indicates better performance.

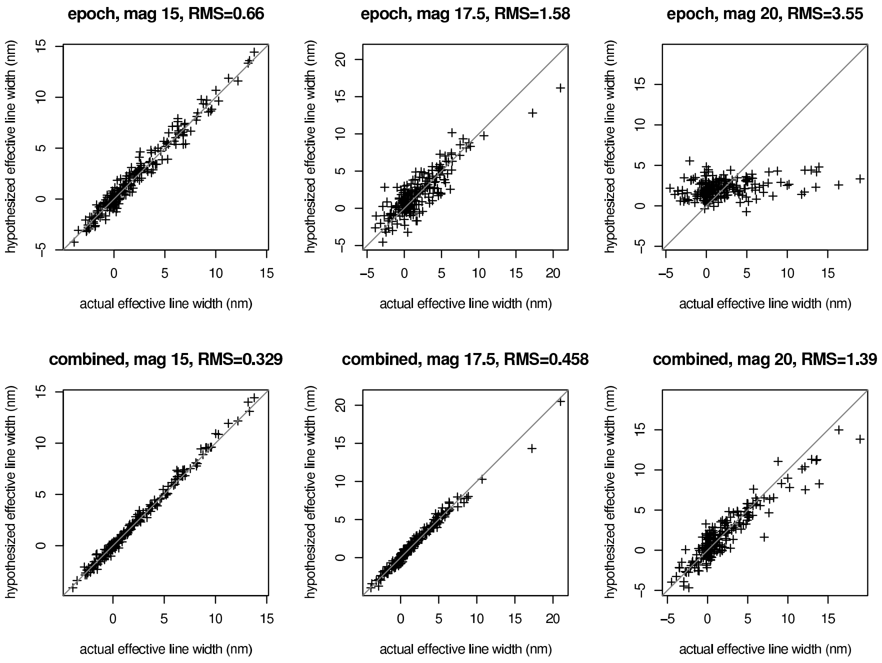

Figure 15: Actual vs predicted H-α effective line width for magnitudes 15 (left column), 17.5 (center column) and 20 (right column), for epoch spectra (top row) and combined spectra (bottom row). Absolute RMS errors are shown in the title of the respective plot.

the algorithm may be extended to other astronomical objects.

The biggest drawback of the the current synthetic model is that it does not include other spectral lines besides the H-α line. Thus, incorporating additional spectral lines is one of the most important improvements that should be made. This requires additional knowledge of astrophysics and was therefore beyond the scope of the current work.

Additionally, validation on real spectra is an important next step in testing the SVR algorithm. Finally, as already mentioned in Section 3.4, the SVR algorithm should be extendable to other spectral lines.

5 Conclusion

The aim of this thesis was to study the possibilities of detecting emission line stars from the BP/RP instrument of the Gaia space telescope using machine learning algorithms. We tried two different approaches and obtained the following results.

Our classification experiments produced reasonable results in detecting H-α emission lines, while illustrating the difficulties of constructing a robust classifier based on limited data in the case of separating Be and WR stars. We discovered relevant discrepancies between the CU8 modelled spectra and local observations of emission line stars.

We have also obtained promising results in using SVR for estimating the effective line width from the low-resolution Gaia BP/RP spectra. To achieve this, we constructed a naive, non-astrophysical model of a stellar spectrum by using PCA on an existing set of stellar continua and combining these with Gaussian approximations of spectral lines. Finally, we proposed several approaches to improve and extend the results obtained, including validating the SVR algorithm on measured spectra, improving our synthetic model with additional spectral lines and attempting to estimate the effective line width of additional spectral lines.

A Resümee

Emissioonijoontega tähtede tuvastamine Gaia kosmoseteleskoobi andmetest
Bakalaureusetöö
Jürgen Jänes
Resümee

Aastal 2011 stardib Euroopa Kosmoseagentuuri (ESA) kosmoseteleskoop Gaia. Teleskoop vaatleb viie aasta jooksul hinnanguliselt miljard objekti, kogudes seeläbi kokku ühe petabaidi jagu andmeid.

Käesoleva bakalaureusetöö eesmärgiks oli uurida, kuidas saab kasutada masinõppe algoritme Gaia kosmoseteleskoobi madalalahutusega BP/RP fotomeetri andmete töötlemisel. Kuna reaalseid andmeid veel ei eksisteeri, siis kasutati töös olemasolevatest kõrge lahutusega spektritest koostatud Gaia vaatluste simulatsioone.

Bakalaureusetöö tulemused saab jagada kaheks osaks. Kõigepealt uurisime tugivektormasinate kasutamist kahe erineva klassifikatsiooniülesande lahendamisel. Esimese ülesandena proovisime eraldada H-α emissiooni sisaldavaid spektreid H-α emissioonita spektritest. Saadud tulemused olid hoolimata piiratud andmetest head. Teise klassifitseerimisülesandena katsusime eraldada Be tähti Wolf-Rayet' tähtedest.

Teise ülesandena üritasime me hinnata tugivektorregressiooni algoritmiga H-α spektrijoone ekvivalentlaiust. Selleks sobiva andmestiku saamiseks kirjutasime me lihtsa emissioonijoontega tähespektri mudeli, mis ignoreerib tegelikkuses spektrit tekitavaid füüsikalisi protsesse. Mudel kasutab peakomponentide analüüsi, et genereerida vaatlustega sarnaseid tähe kontiinuume. Spektrijooned lähendasime Gaussi funktsioonidega. Esialgset tulemused tugivektorregressiooni kasutamisel olid positiivsed.

Tulevikus on plaanis katsetada erinevaid meetodeid klassifitseerimisalgoritmide effektiivsuse parandamiseks, lisada emissioonijoontega tähespektri mudelile astrofüüsikalistest kaalutlustest tuletatud spektrijooni ning uurida tugivektorregressiooni kasutamist teiste spektrijoonte effektiivlaiuse tuvastamisel.

References

[Ano06] Gaia info sheets. pages 1–55, Feb 2006. Available at http://www.rssd.esa.int/index.php?project=Gaia&page=Info_sheets_overview.

[BB03] Jinbo Bi and Kristin Bennett. Regression error characteristic curves. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, 2003.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.

[Bur98] C Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, Jan 1998.

[CBHK02] N Chawla, K Bowyer, L Hall, and W Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, Jan 2002.

[CL01] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[CST04] Nello Cristianini and John Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, 2004.

[DPA] Gaia DPAC. Cu8 cycle 3 simulated data. http://internet1-ci.cnes.fr:8010/simuC3.html.

[Faw04] Tom Fawcett. Roc graphs: Notes and practical considerations for researchers. Mar 2004.

[IZS+08] Y Isasi, I Zaldua, P Sartoretti, X Luri, C Babusiaux, and E Masana. Gog v5.0 user guide. *Gaia DPAC Technical note*, page 35, Nov 2008.

[JLK08] J. Jänes, S. Laur, and I. Kolka. Detection and Characterisation of H-α Emission Lines from Gaia BP/RP Spectra. In *Classification and Discovery in Large Astronomical Surveys*, volume 1082 of *American Institute of Physics Conference Series*, pages 71–82, December 2008.

[KKO+03] Hannu Karttunen, Pekka Kröger, Heikki Oja, Markku Poutanen, and Karl J. Donner, editors. *Fundamental Astronomy.* Springer, fourth edition, 2003.

[MIK+07] Iain Melvin, Eugene Ie, Rui Kuang, Jason Weston, William Noble Stafford, and Christina Leslie. SVM-Fold: a tool for discriminative multiclass protein fold and superfamily recognition. *BMC Bioinformatics*, 8 Suppl 4:S2, Jan 2007.

[MoPGB01] Paul. Murdin and Institute of Physics (Great Britain). *Encyclopedia of astronomy and astrophysics*. Nature Publishing Group, London, New York :, 2001.

[MR09] François Mignard and Sophie Rousset. Gaia data volume in perspective. *Gaia DPAC Newsletter*, (4):4, 2009.

[PTVF92] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1992.

[Ree09] Martin Rees. Pondering Astronomy in 2009. *Science*, 323(5912):309–, 2009.

[Sch06] Peter Schneider. *Extragalactic astronomy and cosmology: an introduction*. Springer, 2006.

[Smi08] K.W Smith. How to use oversampled simulated BPRP data. *Gaia DPAC Technical note*, page 6, Feb 2008.

[SP04] Steven Stahler and Francesco Palla. *The Formation of Stars*. John Wiley & Sons, Chichester, 2004.

[SSBL05] Tobias Sing, Oliver Sander, Niko Beerenwinkel, and Thomas Lengauer. Rocr: visualizing classifier performance in r. *Bioinformatics*, 21(20):3940–3941, 2005.

[TKBJ$^+$07] P Tsalmantza, M Kontizas, C. A. L Bailer-Jones, B Rocca-Volmerange, R Korakitis, E Kontizas, E Livanou, A Dapergolas, I Bellas-Velidis, A Vallenari, and M Fioc. Towards a library of synthetic galaxy spectra and preliminary results of classification and parametrization of unresolved galaxies for gaia. *Astronomy and Astrophysics*, 470:761, Aug 2007.

[ZBI$^+$08] I Zaldua, C Babusiaux, Y Isasi, X Luri, E Masana, and P Sartoretti. Interface control document for GOG v5.0 (cycle5). *Gaia DPAC Technical note*, page 23, Nov 2008.